Aries

Repetitive Strain Injury Prevention Software

Peter Alasdair Kindness

University of Aberdeen

Supervisor: Dr Judith Masthoff

Declaration

I declare that this document and the accompanying code have been composed by myself, and details my own work, unless otherwise acknowledged in the text. This project has not been accepted in any previous application for a degree. All verbatim extracts have been distinguished by quotation marks and all sources of information have been specifically acknowledged.

Signed:

…….

Peter Alasdair Kindness

Date:

…….

Various screenshots and images have been used throughout the making of the Aries software and this report. Any of this images taken from copyrighted software, are owned by the company or person who originally designed and produced the original software.

The use of irreproducible low resolution screenshots to illustrate point(s) in question is believed to qualify as Fair Use under copyright law.

Netbeans is a registered trademark of Oracle Corporation.

Java and MySQL are register trademarks of Sun Microsystems Inc (now owned by Oracle Corporation).

Windows are registered trademarks of Microsoft Corporation.

Macintosh is a registered trademark of Apple Inc.

Qemu is a trademark of Fabrice Bellard.

Acknowledgements

I would like to thank the following people:

- Firstly Judith Masthoff for her advice and support with the writing of this report.
- Secondly a special thanks to Thibaut Lutz, for his extensive knowledge and command of the C language and Linux environment.

Abstract

Whilst working during my industrial placement I became aware of the potential threat of Repetitive Strain Injury (RSI). The company that I worked for had a whole division set up to help people with RSI problems and with the goal of preventing RSI in the first instance. This included setting up any new member's workstations so they are ergonomically correct for that individual combined with offering different peripherals to try and reduce the effects of RSI. Installed on every system was RSI prevention software, it is this that I intend to reproduce and improve on.

The system will offer users a way to prevent RSI by making them take rest breaks or mircobreaks throughout the day. These breaks will force the user to stop typing and will show the user various stretches that can greatly reduce the risk of getting RSI. The system will also offer users advice on how to set up their workstations, information and statistics on how much they have used their computer and allow them to view stretches at anytime.

The goal of my project is to both raise awareness of RSI which affects millions of people worldwide and offer a viable system to help aid in its prevention. The system should educate the user about RSI and ways to prevent it, the analysis phase of the project, as education is still very much the best cure.

This document outlines and discusses the various design, implementation methods and decisions made throughout development, as well as detailing implementation problems and how these problems were overcome.

The system was implemented using the popular platform independent programming language of Java. The system also makes use of other popular technologies such as xml and C++.

The system has received positive feedback from a small group of would be potential users of the system.

Finally this report outlines the future additional work to be implemented to the system, in order for the software to have additional functionality and in order for it to be production ready.

Table of Contents

Declaration 2
Acknowledgements 3
Abstract 4
Table of Figures 11
Table of Tables 12
Glossary of Terms 13
Medical Terms 14
Chapter 1 Introduction 15
1.1 Overview 15
1.2 What is RSI? 15
1.3 Motivation 17
1.4 The Benefit of Intervention 17
1.5 The Challenge 18
1.6 Primary Goals for the System 18
1.6.1 Breaks 18
1.6.2 MircoBreaks 18
1.6.3 Stretches 18
1.6.4 Posture Guide 18
1.6.5 Help Screen 18
1.6.6 Research 18
1.7 Secondary Goals for the System 19
1.7.1 Statistics and Facts 19
1.7.2 Graph and Charts 19
1.7.3 Frequently Asked Questions Screen 19
1.7.4 Intuitive User Interface 19
1.7.5 Robustness 19
1.7.6 Reliable 19
1.7.7 Adherence to Programming Standards 19
1.7.8 Portability 19
1.7.9 Privacy 19
Chapter 2 Analysis 20
2.1 Existing Systems 20
2.2 Payware 20
2.2.1 Break Time 2.0.1 20

2.2.2 Break Reminder 21
2.2.3 Workpace 23
2.2.4 Albion StopNow! 23
2.2.5 Workplace Angel 24
2.2.6 RSI warrior 24
2.3 Freeware 25
2.3.1 RSI Screensaver 1.1 25
2.3.2 EyeDefender 25
2.3.3 Stress Buster 26
2.4 Analysis of Existing Systems 26
2.5 Client Research 27
2.5.1 Questionnaire 27
2.5.2 Focus Group 28
2.6 Outcome of "Client" Research 28
Chapter 3 Requirements 30
3.1 Functional Requirements 30
3.1.1 Product Perspective 30
3.1.2 Model for capturing Requirements 30
3.1.2 Product Functional Requirements 30
3.1.3 General Constraints 41
3.1.4 Assumptions and Dependencies 41
3.2 Database Requirements 41
3.3 Non Functional Requirements 41
3.3.1 User Interfaces 41
3.3.2 Hardware Interfaces 42
3.3.3 Software Interfaces 42
3.3.4 Design Constraints 42
3.3.5 Quality Assurance 42
3.3.6 Memory Requirements 42
3.3.7 Packaging Requirements 42
3.3.8 Legal Requirements 42
3.3.9 Regulatory Requirements 42
3.4 Software System Attributes 43
3.4.1 Reliability 43
3.4.2 Availability 43

3.4.3 Security 43
3.4.4 Maintainability 43
3.4.5 Portability 43
3.4.6 Performance 43
3.4.7 Accessibility 43
3.4.8 Learnability 43
3.4.9 Error Handling 43
Chapter 4 Design and Implementation Decisions 44
4.1 Application Type 44
4.2 Development Language 44
4.3 Architecture 45
4.3.1 Chosen Architecture 45
4.3.2 Storage 46
4.3.3 System Components Interaction 46
4.4 Security 47
4.5 System Protocols 47
4.5.1 Prototype 1.0 47
4.5.2 Prototype 1.1 48
4.5.3 Prototype 1.2 48
4.5.4 Prototype 2.0 49
4.5.5 Prototype 2.1 49
4.5.6 Prototype 3.0 49
4.6 UML Sequence Diagram 50
4.7 User Interface Design 50
4.7.1 Stretches GUI 52
4.7.2 About Screen 55
4.7.3 Help Screen 55
4.7.4 Posture Guide 56
4.7.5 Accessibility 57
4.7.6 Dimensions 57
4.7.7 Navigation 57
4.8 Proposed Database Design 57
4.8.1 Database Schema 58
4.8.1 EER Diagram 58
Chapter 5 Development and Implementation of Core Components 59

5.1 Keystroke logging 59
5.2 Integrating C with Java 61
5.2.1 Define a Native Method 61
5.2.2 Load the Library 61
5.2.3 Create the Main Program 62
5.2.4 Create a stubs file 63
5.2.5 Writing the Native Method Implementation 63
5.2.6 Creating a shared library 63
5.2.6 Running the program 63
5.2.7 Problems/difficulties 63
5.3 Threads 64
5.3.1 Solution to stopping a thread 64
5.4 Displaying all the stretches in categories 65
Chapter 6 Testing 67
6.1 Incremental Testing 67
6.2 User Interface Testing 67
6.3 Stress Testing 69
6.4 User Testing 70
6.4.1 Test Results 70
6.5 Break/Thread Testing 73
6.6 Cross Platform Testing 73
Chapter 7 Evaluation & Future Work 75
7.1 Realisation of Primary Goals 75
7.2 Realisation of Secondary Goals 75
7.3 Evaluation of Completed Work 75
7.3.1 Improvements to core functionality 75
7.3.2 GUI improvements 76
7.3.3 Known Bugs 77
7.4 Future Work 78
Chapter 8 Conclusions 79
References 80
Appendix A - Aries User Manual A1
1 Opening Aries A2
1.1 SplashScreen A2
2 The Aries System A2

2.1 Posture Guide A2
2.2 Aries Overview A4
2.3 Setting the times for the scheduled breaks A4
2.4 Set the Strictness of the Aries system A6
2.5 Rest Breaks A7
2.6 Micro Breaks A9
2.7 Actions A11
2.8 Usage A11
2.9 Stretches A12
2.10 Windows A13
2.11 About Screen A13
2.12 RSI – Frequently Asked Questions A15
2.13 Help A17
2.14 Aries running in the background A18
2.15 Quitting Aries A19
Appendix B - Aries Maintenance Manual B1
1. Requirements and Dependencies B2
2. Building B2
3. Installation guide B2
4. Source Code Structure B3
4.1 Aries.GUI B3
4.2 Aries.Logic B3
4.2.1 Aries.Logic.EventHook B4
4.3 Stretches.Text B4
5. Known Bugs B4
6. Instructions for possible forms of Maintenance B5
6.1 Add a new stretch B5
6.2. Add another Scheduled Break Screen B5
Appendix C - Installing Java SE 6 ..... C1
Linux C1
Windows C1
Further Questions C1
Appendix D - RSI Physical solutions, Techniques and Surgery D1
1. Breaks D1
2. Ergonomics D1

3. Posture D2
4. Lighting D3
5. Ventilation D3
6. Noise D3
7. Mouse Techniques D3
8. Typing Techniques D5
9. Surgery D6
9.1 Carpal Ligament Release Surgery D7
9.2 Trigger Finger Release Surgery D7
9.3 Arthritis Surgery D8
9.4 Surgery Success Rate D8
9.5 Cost of surgery D8
Appendix E - Architecture Research E1
1. Architecture Research E1
1.2. Technical Architecture E2
1.3 System Architecture E2
1.4 Application Architecture E2
1.5 Enterprise Architecture E3
Appendix F - Focus Group Plan F1
1. Introduction F1
2. Discussion Guidelines: F1
3. Introductory Questions F1
4. Knowledge of RSI F1
5. Current solutions to RSI F2
6. Aries F2
7. Closure of session F2
Appendix G - Focus Group Questions G1
1. General RSI Related Questions G1
2. Software Currently Used and Daily Routine G2
3. Evaluation of “RSI Prevent” proposed features G2
Appendix H - Questionnaire H1

Table of Figures

Figure 1- Diagram of Hand 16
Figure 2 - Break Time Main Window 21
Figure 3 - Break Reminder's settings window 22
Figure 4 - Workpace's graphical representation of user's activity 23
Figure 5 - Albion StopNow Interface 24
Figure 6 - Workplace Angel settings screen 24
Figure 7 - Stress Buster Short Pause screen 26
Figure 8 - The convention stages of requirements capture 30
Figure 9 - UML use case diagram detailing primary functionalities 34
Figure 10 - Three Tier Architecture of System 46
Figure 11- UML Sequence Diagram illustrating the basic interactions between the user and the system 50
Figure 12 - RSI Warrior's GUI 51
Figure 13 - Workpace menus and GUI 51
Figure 14- Aries Main GUI 52
Figure 15 - RSI Warrior Stretch Player 52
Figure 16 - Workpace Break 53
Figure 17 - Aries Rest Break Window 53
Figure 18 - Break Reminder, Workpace and Big Stretch 54
Figure 19 - Aries MicroBreak 54
Figure 20 - Aries Scheduled Break Screens 55
Figure 21 - Aries About screen 55
Figure 22 - Aries Help Screen 56
Figure 23 - Aries Posture Guide 56
Figure 24 - Database EER diagram 58
Figure 25 - Event Listener Hook written in C 60

Table of Tables

Table 1- Functional Requirements Table 33
Table 2 - System Monitor Use Case 35
Table 3 - System Timer Use Case 36
Table 4 - Posture Guide Use Case 37
Table 5 - MicroBreak Use Case 37
Table 6 - Rest Break Use Case 38
Table 7 - Read XML Use Case 39
Table 8 - Display All Stretches Use Case 39
Table 9 - Schedule Break Use Case 40
Table 10 - Display User Actions Use Case 41
Table 11 - Comparison of Object Oriented Programming Languages 45
Table 12 - Prototype 1.0 48
Table 13 - Prototype 1.1 48
Table 14 - Prototype 1.2 48
Table 15 - Prototype 2.0 49
Table 16 - Prototype 2.1 49
Table 17 - Prototype 3.0 49
Table 18 - Window Dimensions 57
Table 19 - Interface Testing Results 69
Table 20- Known GUI Issues 77
Table 21 - Known Bugs 77

Glossary of Terms

C	Programming Language invented in 1971 which forms the basis of many modern programming languages
C++	Object oriented C
CTS	Carpal Tunnel Syndrome
CVS	Computer Vision Syndrome
FAQ	Frequently Asked Questions
GUI	Graphical User Interface
IDE	Integrated Development Environment – provides complete set of tools for programming
IASA	The International Association of Software Architects
Java	Object orientated programming language developed by Sun Microsystems
Keylogger	A virus that listens to the properties of the keys the user presses before passing the information onto a third party. This is a form of getting sensitive information such as passwords.
JVM	Java Virtual Machine
JXGrabKey	A Java native implementation (JNI) library for easy use of global X11 hotkeys on Linux from java. It was inspired by the JIntellitype project. JXGrabKey was originally created as a subproject of Coopnet
Linux	Open Source Operating System
Mozilla Firefox	Mozilla Firefox is a free and open source web browser descended from the Mozilla Application Suite and managed by Mozilla Corporation.
OOS	Occupational Overuse Syndrome
OS	Operating System
QEMU	A generic and open source machine emulator and virtualiser.
RSDS	Reflex Sympathetic Dystrophy Syndrome
RSI	Repetitive Strain injury
SVN	Subversion – a versioning control system
Swing	Java framework for building Graphical User Interfaces
TOS	Thoracic outlet syndrome
UNIX	Family of Operating Systems which implement the UNIX specification
UNIX Kernel	The core or key components of the operating system — consisting of many kernel subsystems like process management, memory management, file management, device management and network management.
Windows	Microsoft's desktop and server Operating System
XML	Extensible Markup Language

Medical Terms

Term	Description
Causalgia	Intense burning pain and sensitivity to the slightest vibration or touch, usually in the hand or foot, at a site some distance removed from a wound that has healed. This phenomenon was first described in 1872 by the American neurologist Silas Weir Mitchell (1829-1914).
DeQuervain's Syndrome	*de Quervain's tenosynovitis* is a condition that causes pain on the inside of the wrist and forearm just above the thumb.
Intersection Syndrome	Painful condition that affects the thumb side of the forearm and occurs where the two muscles intersect or cross over the two underlying wrist tendons.
Reflex Sympathetic Dystrophy Syndrome	A chronic condition that is characterized by severe burning pain with extreme sensitivity to the touch, pathological changes in bone and skin, excessive sweating and tissue swelling.
Stenosis	An abnormal narrowing or contraction of a duct or canal
Synovium	See Tenosynovitis
Thoracic outlet syndrome (TOS)	A broad term that refers to compression of the neurovascular structures in the area just above the first rib and behind the clavicle that results in upper extremity symptoms.
Tendonitis	The inflammation (swelling) of a tendon
Tenosynovitis	The inflammation of the sheath that surrounds a tendon. (The sheath is called the synovium.)
Trigger finger/thumb	A form of stenosing Tenosynovitis

Chapter 1
Introduction

This section of the report will introduce the concept behind my project as well as outline its goals and aims. The objectives and motivation behind such a project will also be addressed.

1.1 Overview

The aim of this project is to design and develop a software package called Aries.
Aries is a practical and educational Repetitive Strain Injury (RSI) prevention system. The system will monitor a user's activity and will force the user to take breaks throughout the working session whilst educating them about how to prevent RSI.

The system has been designed to divide up the user's pattern of work, as research has shown that even a break of a few seconds can dramatically reduce the chance of RSI developing. The system should also give the user the opportunity to look at various RSI preventing stretches at anytime, set the time between breaks and view various statistics about their activities during that session.

The system offers two forms of breaks. The first of these are the main breaks that appear throughout the users session at pre determined periods of time. These so called rest breaks can last for a pre determined time, usually ten minutes. The second of these breaks is a MicroBreak. These MicroBreaks only last for a few seconds and are determined by the user work intensity. Although the primary aim of the system is to prevent RSI, the system also caters for users that are already currently experiencing some of the affects of the syndrome.

1.2 What is RSI?

Repetitive Strain Injury (RSI) is a pain syndrome that affects millions of people world-wide and in 2006 nearly half a million people suffered from some form of RSI in the UK [1]
In a more recent survey, 68% per cent of the UK's workforce suffered from RSI, with the most common areas effected being the back, shoulders, wrists, and hands [2].

With 5.4 million working days being lost by sick leave as a result of RSI on average every year [3] with an estimated cost to UK industry in the range of five billion pounds to twenty billion pounds annually [4], RSI is a very real condition and one that both employers and individuals should become more aware of.

Employers are indeed becoming more aware of the threats that RSI presents to their workforce and their need to take measures to prevent RSI. This was highlighted in 2002, when Barclays Bank had to pay compensation to the sum of £244,000 to a former employee who was forced to give up work because of pain in her right hand [10], a pain that was a direct result of RSI.

RSI is a broad syndrome that encompasses a variety of musculoskeletal disorders. These can affect tendons, muscles and nerves in the hands, arms and upper back.

RSI is caused when muscles are kept in a tense state for prolonged periods of time, due to either poor posture or repetitive motions.

The most common types of RSI related disorders which can develop are:

- Carpal tunnel syndrome (CTS) is the most common type of RSI disorder diagnosed and occurs when the median nerve is compressed by the surrounding sheath encompassing the tendons at the wrist. CTS can be indicated by symptoms such as burning and tingling, pain, coldness, and sometimes weakness in parts of the hand which grow gradually.[11]

- Tendonitis, which is the inflammation of the tendons. [12]

- Tenosynovitis is the inflammation of the fluid-filled sheath surrounding a tendon (the synovium). It is the most common cause of tendonitis and is closely related to stenosing tenosynovitis. [12]

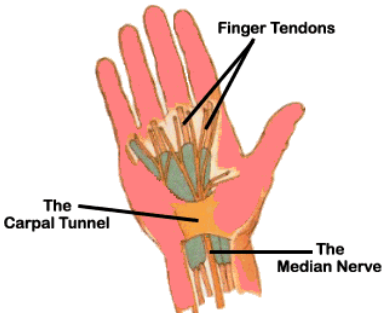


Figure 1- Diagram of Hand

- DeQuervain's Syndrome is an inflammation of the synovium or carpal tunnel that surrounds the two tendons which control the movement of the thumb. This is often suffered by typists as it is caused by the thumbs repetitive up-down motion required to operate a keyboard. Difficulty gripping objects, pain and swelling over the thumb side of the wrist and numbness on the back of the thumb are the usual symptoms of this disorder. [13, 14]

- Thoracic outlet syndrome (TOS) consists of a group of distinct disorders that affect the nerves that pass into the arms from the neck as well as various blood vessels and nerves between the base of the neck and armpit. These disorders vary and have little in common except for the area of occurrence.[15]

- Trigger finger/thumb is a type of stenosing tenosynovitis in which the sheath around a tendon of a finger or thumb becomes inflamed and restricts the movement. The result of this is that the affected joints may become painful to straighten once bent and make a cracking noise when flexed.[16]

- Intersection Syndrome is a painful condition that affects the thumb side of the forearm and occurs where the two muscles intersect or cross over the two underlying wrist tendons. Intersection Syndrome is often confused with DeQuervain's syndrome. [17]

- Reflex sympathetic dystrophy syndrome (RSDS) also known as complex regional pain syndrome. This is a chronic condition that is characterized by severe burning pain with extreme sensitivity to the touch, pathological changes in bone and skin, excessive sweating and tissue swelling. RSDS is currently classified as a variant of causalgia, rather than being caused by trauma. [18]

After any of these conditions have been diagnosed a trained physician usually prescribes a specific therapy designed for that condition. However in most situations these therapies have little or no effect. The best cure still is prevention.

1.3 Motivation

During an industrial placement in a professional environment I became aware of the potential threat of RSI. Several members in the Information Technology division suffered from one or more syndromes that are associated with RSI. In this new age of technology, computers have been become deeper and more firmly entrenched within our society. With roughly five hundred million computers in the world and with numbers of users reaching close to one billion, the computer has influenced and brought about a change in lifestyle to our lives and as a by product, RSI is becoming an increasingly real threat to the future generations.
In the light of this it is my goal to design and implement a system that will help educate users of RSI whilst preventing the condition from occurring in the first instance.

Results gathered from consumer research have highlighted the issue of lack of education on the issue of RSI as being rife amongst younger users. Although my system's primary aims are to help prevent and reduce the affects of RSI, its secondary goal is to educate and inform users on the very real threat of RSI.

1.4 The Benefit of Intervention

Research carried out in the US discovered that for every dollar that was invested into RSI prevention techniques in the workplace, through the mediums of ergonomic improvements to the workers workstation, there was a return to the employer of $17.80 [5]. Organisations that have already developed and put into place such strategies have found that RSI symptoms were three times less likely to occur [6] and as a result benefited through their employee's loss of work. Not only are such strategies of benefit to the employer, research has shown that employees feel more

respected and appreciated resulting in increased skill development and job satisfaction [7].

1.5 The Challenge

In order to fulfil these goals and create a system that users would find most beneficial and not a hindrance, I had to research the pros and cons from existing systems in the market and look for the best technical solutions to potential problems.

1.6 Primary Goals for the System

In order to combat RSI the proposed system would need to offer similar functions and fulfil the following key issues.

1.6.1 Breaks

The system will present the users with a window, after a predetermined period of time, displaying a series of RSI prevention stretches. These breaks will last for 5 minutes.

1.6.2 MircoBreaks

The system will lock the users screen for a few seconds. Research shows that taking your hands off the mouse or keyboard even for a few seconds can greatly reduce the risk of RSI developing. The system will present the user with a window that will have a countdown displayed upon it; these MircoBreaks will be initiated in correspondence to the users work intensity.

1.6.3 Stretches

The system should be able to provide the user with the ability to look through the systems stretches at any time.

1.6.4 Posture Guide

The system will present the user with a Posture guide on start up. This guide will allow the user to adjust their workstation in order to make it more ergonomically efficient. This screen should be able to be viewed at anytime.

1.6.5 Help Screen

The system will have help capabilities. This means that if a user is unsure about any of the systems features, they will be able to find instructions.

1.6.6 Research

Extensive research will be carried out throughout the duration of the project in order to ensure the best approach has been taken and all possible avenues have been explored.

1.7 Secondary Goals for the System

With the realisation of such issues the proposed system should also look into offering the user further functionality and adhering to design protocol such as:

1.7.1 Statistics and Facts

The user should be able to view various statistics on their behaviour during the session. The statistics will include things such as keys pressed, mouse distance travelled and programs used.

1.7.2 Graph and Charts

The user statistics should be presented in various charts and graphs.

1.7.3 Frequently Asked Questions Screen

The System will offer FAQ's and their answers about RSI to the user, to help increase the user's knowledge and understanding of the threats of RSI.

1.7.4 Intuitive User Interface

The system will have a simple, easy to use graphical user interface

1.7.5 Robustness

The system must be robust in order to deal with improper user inputs and general abuse to the system without causing the system to crash or hang. Fuzz testing will be carried out on the system to insure that the system is robust.

1.7.6 Reliable

The system must be reliable in order to perform and fulfil its requirements every time the program is run.

1.7.7 Adherence to Programming Standards

The system must be programmed with adherence to programming standards. This is in order to increase maintainability and can be achieved using appropriate comments and formatting.

1.7.8 Portability

The system must be portable, in order to be accessible to as wide a consumer base as possible. It must be able to run on the most widespread Operating Systems (OS). These are: Windows, Linux based systems and Apple Macintosh.

1.7.9 Privacy

The system should also be aware of the user privacy and must be careful with saving and dealing with monitoring the user's activities.

Chapter 2
Analysis

At the beginning of this project research was undertaken to determine the feasibility of such a project and to establish the existing features of current system currently in the market place.

Initial research uncovered that there were many existing systems offering a similar system. The following chapter details these findings.

2.1 Existing Systems

As mentioned earlier, it was after I encountered a similar RSI system to the one I proposed for my project, that I became aware of the potential threats of RSI and had the idea to reproduce and improve upon such a system.

Upon researching similar RSI prevention and reduction software I became aware that there are many currently in the market. The available systems cover a wide spectrum of RSI relating syndromes and range from being specifically designed to combat one syndrome to a wide array of syndromes.

I also looked in to the classification of software that the current systems fall under. This was to research the potential scope and financial gain that could be achieved by a project like mine. The software that I researched tended to fall into the two categories. The first being the traditional form of software where a product has been designed with the intention to sell to buying consumers in order to make a profit out of the investment in the production of such a system. This is commonly referred to as Payware, The other form of software out there is that of Freeware, this when the software is available for use at no cost or for an optional fee.

2.2 Payware

2.2.1 Break Time 2.0.1

Break Time [20] is a rest break reminder system designed to prevent RSI, Computer Vision Syndrome (CVS) and Occupational Overuse Syndrome (OOS). Break Time's designers believe that reminding the user to take regular breaks in order to relax and exercise throughout the day will prevent weariness and fatigue. They also believes that users should listen to quiet music during these breaks in order to enhance the relaxation. As a result of this, Break Time provides the user with calm and soothing music in the background during their break. The system monitors the user's time on the computer and will recommend times for the user to take breaks in order to reduce the affects of RSI. The system main interface is shown below in figure 2.

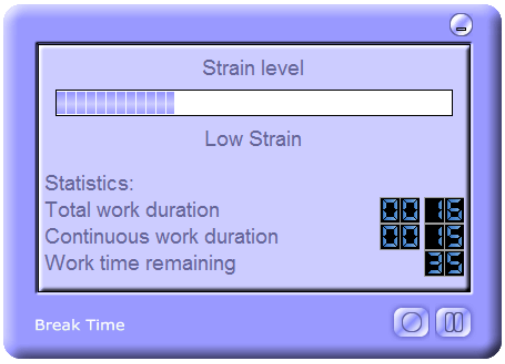


Figure 2 - Break Time Main Window

Break Time offers the following features:

- Rest break reminders.
- Monitoring of the user's activity.
- Customizable Information Window with a strain indicator.
- Locks the users screen during breaks to enforce them.
- Provides three types of reminders. These include: a message in the Main Window/Information Window/Lock Screen.
- Allows the user to start a break at any time.
- Allows the user to postpone or ignore breaks.
- Offers a pre-break warning to users (a flashing tray icon and a warning sound)
- Plays a sound to inform the user of the start and end of the break
- Plays music during the breaks with the following formats being supported: MID, WAV, MP3 and OGG formats.
- The systems appearance can be altered by the user with various different skins being available
- The system also boasts a multi-language user interface

2.2.2 Break Reminder

Break Reminder [20] is a software package designed to both prevent and assist convalescence to RSI. Break Reminder has a small program interface that runs in the background and monitors computer use and reminds, or forces, the user to take a break depending upon their settings. The system's settings screen is shown below in figure 3.

Break Reminder boasts the fact that is highly configurable to each individual user's circumstances, whether they are an OOS sufferer experiencing symptoms or if they want to prevent RSI.

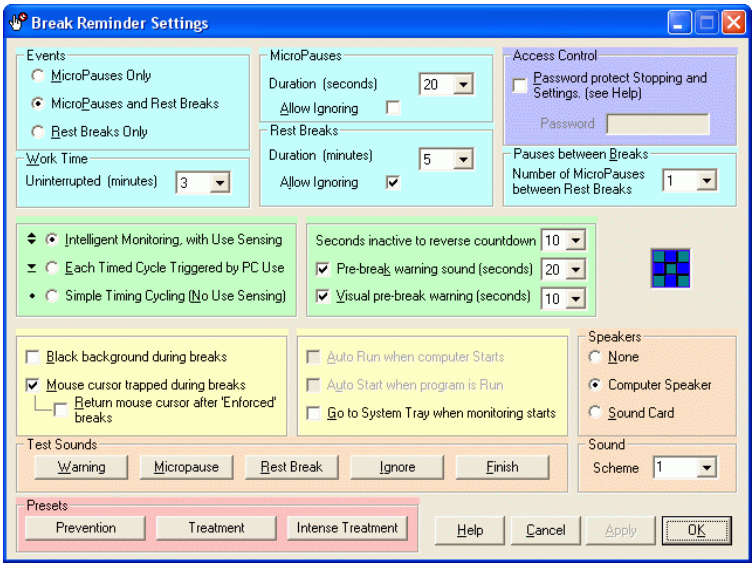


Figure 3 - Break Reminder's settings window

Break Reminder offers the following features:

- Mini pauses, whose duration is recorded in seconds
- Rest Breaks, whose duration is recorded in minutes
- Offers the option to "Black Out" the whole screen during breaks and pauses.
- Allows the user the option to either ignore a break or not depending on pre determined settings
- Allows the user to choose the number of Mini breaks between Rest Breaks
- Boasts three operational modes:
    - Basic timed reminders – set times
    - Key/Mouse sensing – computer use influencing break times.
    - Key/Mouse triggering of each successive timed period. (resumption of timing on return to PC use)
- Offers three preset configurations, these consisting of: RSI prevention. RSI sufferer or intense RSI sufferer.
- The system allows the user to pause the break reminder at any time without restarting timers and counter (manage interruptions)
- Supports password protection option for the settings screen.
- The system has an auto start-up function, meaning it starts every time the computer has been booted.
- The system records the user activities.
- The system supports sound and has sound options for indicating the start and end of breaks.

2.2.3 Workpace

Wellnomics® Workpace [21] claims to be the only scientifically validated RSI prevention software. Workpace (shown in figure 4) has had growing success in the commercial sector as it gets people to accept personal responsibility for their own safety. This protects companies from compensation claims and high insurance costs. Like my proposed system it educates computer users in practical ways to prevent computer fatigue, discomfort and injury. As a result of this it intends to ensure happier, healthier employees meaning increased productivity.

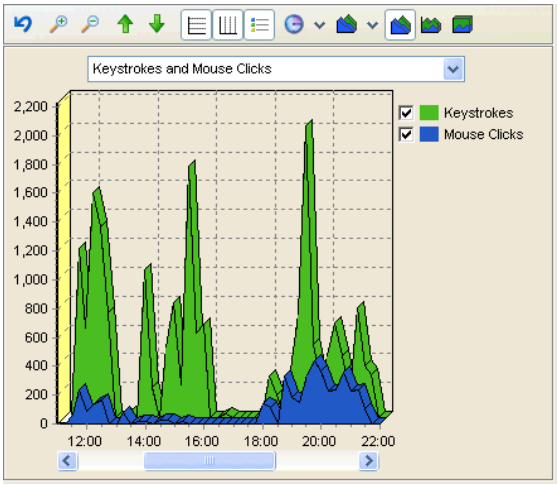


Figure 4 - Workpace's graphical representation of user's activity

Workpace looks to help the user achieve safe and healthy work habits by adhering to the following five principles:

- Workpace looks to educate the user about muscles fatigue and recovery from such fatigue
- The system alerts the user when they need to take breaks
- It aims to guide the user through regular fatigue, and as a result RSI, reducing exercises and stretches.
- It also monitors the user's work intensity throughout the sessions and
- Provides the user with feedback in the form of graphs and charts.

2.2.4 Albion StopNow!

This software is designed to remind the user to take RSI preventing breaks. StopNow's [22] (shown in figure 5) reminder is designed to be:

- Obvious to the user, but not obvious to anyone else
- They are actually designed to be "mildly annoying" in order to actually make you listen and take a break.
- Designed to be appropriate for a work environment, meaning that there are no pop-ups that may result in lost key strokes.

Figure 5 - Albion StopNow Interface

2.2.5 Workplace Angel

Workplace Angel [23] is designed to be a flexible and easy to use program that helps in setting up the user's workstation for comfortable and healthy work. The program also reminds the user to take pauses throughout the working day and demonstrates animated exercises that are designed to help prevent RSI on demand.

Figure 6 - Workplace Angel settings screen

2.2.6 RSI warrior

RSI warrior [24] is similar to the Workpace system described earlier. It actively looks to prevent RSI from occurring as well as help users that already suffer from RSI.

RSI warrior boats the following features:

- Rest break reminders that are timed based on how hard the user is actually working on the computer, whilst being as unobtrusive as possible.
- RSI warrior boasts to have over twenty professionally designed stretches that are shown during the rest breaks. The stretches are thirty seconds long in duration and guide the user through the stretches with the help of smooth 3D animations.

- The system also has an ergonomics wizard that is designed to show the user how to properly configure their workstation.
- Posture reminders are also shown to remind the user to correct their seating posture
- The system boasts a new initiative concept in automatic clicking. This is designed to get rid of the strain associated by constantly clicking the mouse. This works by automatically issuing a click whenever the mouse is held stationary for a short period of time. Although this takes some getting used to, it is great idea for people who suffer from mouse clicking related forms of RSI.
- The system also offers tracking and reporting of key work statistics that are designed to help the user understand how they use their computer.

2.3 Freeware

2.3.1 RSI Screensaver 1.1

This software package offers the user various stretches instead of the usual desktop screensavers. This software package is effectively just a series of pictures showing RSI prevention stretches that the system then sets as the users screen saver. There needs to be a prolonged period of inactivity however for the screensaver to be activated. This means the user will either not be at their workstation any more or that the user will continue working until they begin to feel the effects of RSI, which by then could be too late.

2.3.2 EyeDefender

The EyeDefender [25] package is software package specially design to focus on the eyes of the user. The system has been designed to prevent Computer Vision Syndrome (CVS) that can result from working on a computer for prolonged periods of time. CVS symptoms include eyestrain, blurred vision, delayed focusing, headaches, back and neck aches, dry and burning eyes, light sensitivity, distorted colour vision, forehead heaviness, and sore eyes [1].
EyeDefender boasts the following features:

- Allows the user to control the level of eye strain.
- Allows the user to adjust the time period for a popup reminder.
- Allows the user to set up the duration of the rest break.
- Allows the user to have a break at any time.
- The counter between breaks can be reset at any time.
- The user can stop the counter when you do not want to be interrupted
- The counter can be set to stop automatically after some time of inactivity
- The system also boasts four rest modes: slideshow, visual training, screensaver launch, or popup tray reminder
- The system also starts when your computer is switched on.

2.3.3 Stress Buster

This RSI prevention software was designed to prevent upper limb disorders such as back pain, neck, wrists and shoulder pains, carpel tunnel syndrome, tenosynovitis and tendinitis. Like my proposed system it aims to help aid the recovery from RSI as well as the prevention. Stress Buster [26] runs in the background and reminds the user to takes breaks throughout the day in order to help the prevention of RSI.

Stress Buster boasts the following features:

- The system offers the user three levels of breaks. These are Short Pause, Short Break and Long Break.
- The system offers the user to adjust the times for the breaks.
- The systems monitors the users activity and advises the user when a break is recommended.
- The system allows users to ignore or delays breaks (this feature can also be disabled by the user)
- The system also has the option to lock the screen during breaks to enforce them.
- Break reminder sounds are customisable.
- Stress Buster does not record any of the user's computer activities. Thus meaning the system cannot be used as a means to spy.
- Can be set to run on windows start-up.

Figure 7 - Stress Buster Short Pause screen

2.4 Analysis of Existing Systems

Table 1 below shows the comparison of the existing systems researched to that of the proposed functions for the Aries system. From studying the table it is clear to see that there is no system currently available that would offer the proposed functionality of the Aries system and that any system that supports many of Aries features are Payware.

System	Distribution Type	Rest Break	Micro Break	Monitor User Activity	Intensity	Scheduled Breaks	Sounds	Settings	Stretches
Breaktime	Payware	✔		✔	✔		✔		✔
Break Reminder	Payware	✔	✔	✔				✔	✔
Workpace	Payware	✔	✔	✔	✔			✔	✔
Albion StopNow!	Payware			✔	✔			✔	✔
Workplace Angel	Payware	✔	✔				✔		✔
RSI Warrior	Payware	✔		✔	✔			✔	✔
RSI Screensaver	Freeware								✔
Eye Defender	Freeware	✔		✔					
Stress Buster	Freeware	✔					✔	✔	

Table 1 - Existing Systems Comparison Table

2.5 Client Research

Before development on the proposed system could begin, further research was required to establish the following:

- What the potential consumer base knew of the threats of RSI?
- What their daily routine and behaviour currently is?
- Which features of the proposed software they would find most useful?
- Are there any additional features that consumers would like to see included that may have been overlooked or unthought-of?
- What user interface requirements the users would be looking for?

In order to answer these questions I decided to carry out two forms of "client" research. These consisted of a questionnaire and a focus group.

2.5.1 Questionnaire

The questionnaire was designed to be able to access a wide range of the potential client base and consisted of broad RSI and current computer behaviour questions. All the questionnaires were fully anonymous and engineered to give an insight as to what current computer users feel and experience.

The template for the questionnaire used, can be found in Appendix H. For samples of completed questionnaires please send your enquires to p.kindness.06@abderdeen.ac.uk.

2.5.1.1 Questionnaire Results

A brief overview of the questionnaire results can be seen below in Table 2.

Age	<16	16-24	25-34	35-44	45-54	55+
	1	5	2	1	1	0
Gender	Male		Female		Other	
	8		2		0	
Average Hours on a computer per day	<1	2-3	4-6	7-8	9+	
	0	2	5	3	0	
Had RSI problems before	Yes		No			
	4		6			
Primary use of computer	Typing	Spreadsheets	Browsing Internet	Gaming	Emails	Monitoring systems	Other
	3	1	4	1	1	0	0
What do you use your computer for	Work		Leisure			
	6		4			
Longer or shorter breaks	Longer		Shorter			
	8		2			
Is your output measured	Yes		No			
	2		8			
Workstation well designed	Yes		No			
	2		8			

Table 2 - Questionnaire Results (sample of 10)

2.5.2 Focus Group

The focus group was held with Hampton Associates Web Development team in their office. The focus group was designed to get an insight in to the daily routine of a professional office and to see whether they were aware of the potential threats of RSI. The focus group also discussed where such a system would be welcomed into a working environment and whether there were any extra features that they might want incorporated into the system (see Appendix F).

Details of the results of the focus group can be found in Appendix G.

2.6 Outcome of "Client" Research

The outcome of the questionnaires and focus groups highlighted the fact that although RSI was known about by most professionals and older people, mainly because they had experienced it before in some shape or form, the younger participants were unaware of the potential threats that it can inflict to one's health. And although many had heard of hardware solutions to RSI few had heard of software solutions. It is these outcomes that I wish to address with my proposed system, to offer a software solution to RSI and to educate users about the potential threat of it and how to take actions to prevent it.

With regards to the functionality of an RSI prevention system, results seemed to be conclusive that longer few breaks were preferred over shorter more frequency breaks and that although sounds would be a nice option to have, they were more than likely be turned off. Feedback on Aries' proposed features where positive and the idea of having RSI preventing stretches being a key feature, providing that they were subtle and not distracting to others in the work place.

Chapter 3
Requirements

This chapter defines the needs and goals for the Aries system, detailing both the functional and non-functional requirements that the implementation must adhere to.

3.1 Functional Requirements

3.1.1 Product Perspective

In a nutshell my proposed system should provide the user with a series of RSI prevention-helping stretches that pop-up after a predetermined time of activity.

3.1.2 Model for capturing Requirements

Although the users understanding of the problem is subjective, the problem itself in reality is ambiguous. This is due to the fact that the users are the "expert" in what the system needs to do and the developers only link to the problem. Due to the fact that the users understanding of the problem is not perfect (note the understanding "cloud" in Figure 8 below) the developer needs to scrutinise and refine the requirements. When the preliminary requirements are defined in the Requirements definition the analyst must rewrite and eradicate any ambiguity before writing the Requirements Specification. This forms the contract between the user and the software's developers. Once this is achieved it is the job of the analyst to specify a system that can satisfy these requirements. This details both the functional and non-functional requirements of the system. These requirements are defined here.

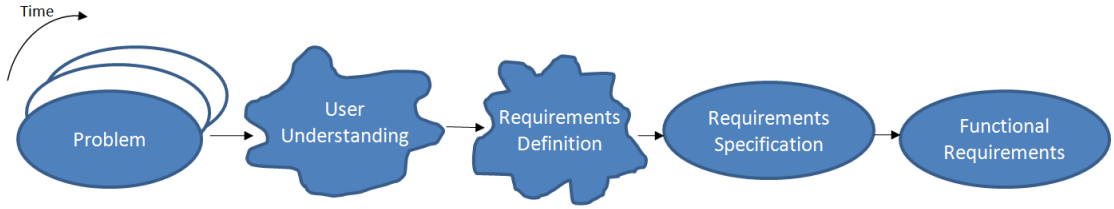


Figure 8 - The convention stages of requirements capture

3.1.2 Product Functional Requirements

The functional requirements for my system were devised using a combination of the results gathered from the focus group and questionnaires used in unison with my own personal experience and knowledge on the matter. This experience has been established by using similar systems to the one proposed and through extensive research.

3.1.2.1 Description of Actors

In this section the following actors are referred to:

U: The user using the system.

Sys: The system itself.

3.1.2.2 Textual List of Required Functionality

The following table (Table 3) details the required functionality for the Aries system. The actor that interacts with each of the requirements is specified in the 'Actor' column and the system version that that requirement is needed for is specified in the 'Required for Prototype' column. The 'Dependencies' column highlights and pre-conditions that the requirement may have, in other words, what requires need to be fulfilled prior to the current requirement being able to be implemented.

U = User, Sys = System

No	ACTOR	REQUIREMENT DESCRIPTION	DEPENDENCIES	REQUIRED FOR PROTOTYPE
10	Sys	System shall be able to tell the user how many keys that have been pressed during the present session		1
15	Sys	System shall be able to tell the user how far the cursor have travelled in centimetres		1
20	Sys	System shall be able to allow the user to see stretches at any time These stretches will cover areas such as Legs, Back, Neck, Hands and Eyes.		1
30	U	User shall be able to exit the system at anytime		1
40	U	User shall be able to see information about how to use the system at anytime. This will be in the form of a help screen.		1
45	U	The user shall be able to see information about the system. An About screen for displaying why the program was made.		1
50	U	User shall be able to set the duration of the main breaks (2,6,10 minutes)		3
55	U	User shall be able to set the duration of the micro breaks (5, 15, 25 seconds)		3
60	Sys	System shall be able to tell the user how intensely they are		2

		working: This will be calculated by how many keys have been pressed during a time period, such as 5 minutes Intensity will be measured on a scale of 5.		
65	Sys	System shall be able to show the user how long it is until the next break		2
68	U	User shall be able to set the strictness of the breaks	50, 55	3
70	Sys	System shall be able to determine when the next micro break shall be depending upon intensity level.	55	3
74	U	User shall be able to set the duration of uninterrupted work		3
75	Sys	System shall be able to show the overall time (in hrs:mins) that the system has been in use (Total time the user is on the computer)		2
80	Sys	System shall be able to allow the user choose where they are allowed to ignore breaks or not		3
82	U	User shall be able to ignore a break (if non-compulsory)	80	2
85	Sys	System shall be able to Show the user what programs they have been using from the present session	75	3
90	Sys	System shall be able to tell the user how many keys their left hand has pressed	10	3
91	Sys	System shall be able to tell the user how many keys their right hand has pressed	10	3
100	Sys	The system's breaks will always appear on top of any other window that the user may have open		3
105	U	User shall be able to have the system blacken the background during breaks		3
110	U	User shall be able to disable the		3

		mouse during breaks		
120	Sys	The system should have a pre-break sound		3
121	Sys	The system should have a post-break sound		3
123	Sys	The system should have a pre-break visual		3
130	Sys	The system should have password protected settings		4
135	Sys	The system should give the user the option to show the correct posture guide on start-up		4
136	Sys	The system should show posture guide on start-up (unless unselected)	135	4
137	Sys	If unselected, system should not show posture guide on start-up	135	2
140	Sys	The system should allow the user to set the frequency of breaks for the user by selecting from the choices of: RSI prevention, Got RSI or Bad RSI		4
150	U	The user should be able to set a time for when the system will alert them of when it is time for the morning break		2
151	U	The user should be able to set a time for when the system will alert them of when it is time for the lunch break		2
152	U	The user should be able to set a time for when the system will alert them of when it is the end of the day		2

Table 3- Functional Requirements Table

3.1.2.3. UML Use Case Diagram

Use case for Primary Functionalities.

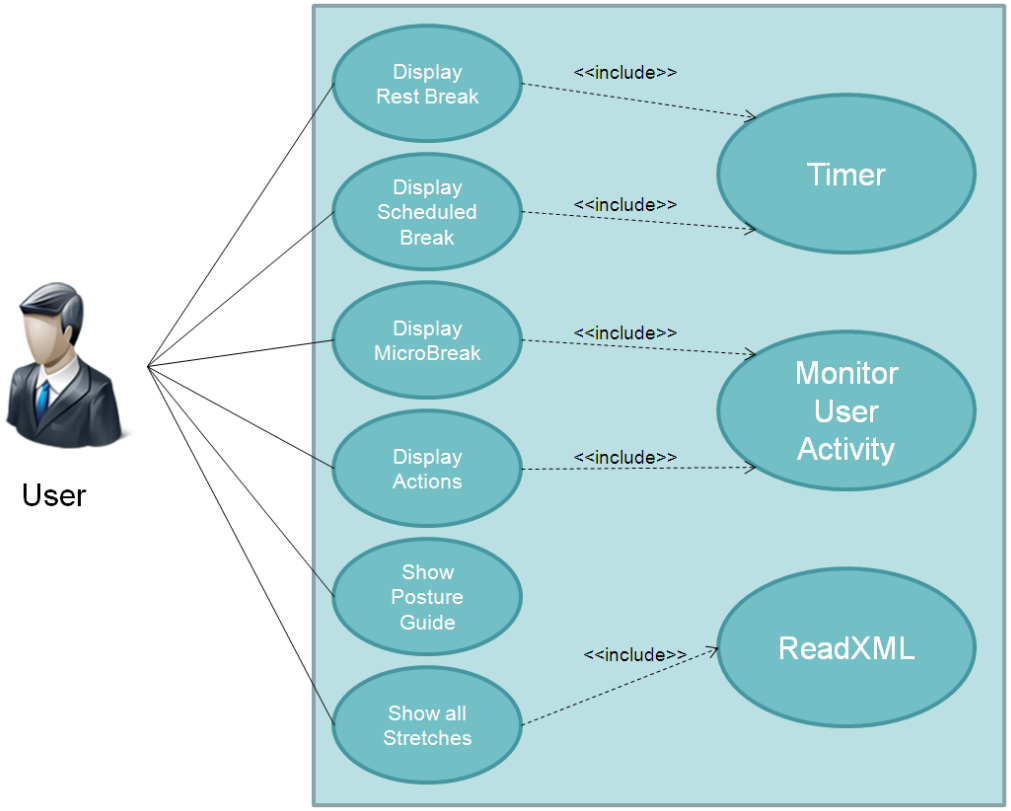


Figure 9 - UML use case diagram detailing primary functionalities

The definition of the Use Case for the system monitoring the user's activities during the session is detailed here in Table 4.

System Monitors User	
Use Case Name	Monitor User Activity
Scope	Aries
Level	System Goal
Primary Actor	None
Stakeholder & Interests	
User	Must not notice a drain or reduction in systems performance
	Must not be distracted from their usual work
	Differentiate between mouse clicks and key presses
	Be able to see statistics on their computer use, such as: Overall computer usage, total number of mouse click, total number of keys pressed and the total distance that the mouse (cursor) has travelled in centimetres
	The system must not throw any errors whilst performing the monitoring
Preconditions	Application is open

	The system has the ability to listen to the users actions
	The users work consists of the use of peripherals such as a mouse and keyboard
Success Guarantee	The system listens to the users actions by counting the number of mouse clicks, the number of key presses and the distance the mouse has travelled
Main Success Scenarios	
1)	User opens the Aries Application
2)	User Events, such as key presses and mouse clicks are displayed in the system's "Usage" tab
Extensions	
a)	If the system is cannot listen to the users activities
i)	Appropriate error message displayed
ii)	User clicks ok and the system does not hang or crash
b)	At any time the user should be able to see their work intensity level
i)	Statistics gained from these observations will be used to determine the users work intensity level.
ii)	A MircoBreak will appear if the user's intensity level is above 3. (out of 5 levels)
c)	At any time the user wants to exit the application
i)	User clicks on the "Exit" button in the "File" pull down menu
ii)	User "right-clicks" on the Aries icon in the systems notification area and click "Exit" from the pop-up menu
Special Requirements	Must not restrict the user's normal activities
	Must not consume much of the CPU's processing power

Table 4 - System Monitor Use Case

The Use Case for the systems timer is detailed in Table 5 below. The Timer deals with all the time management within the system, such as the system time and threads.

Handles the systems time	
Use Case Name	Timer
Scope	Aries
Level	System Goal
Primary Actor	None
Stakeholder & Interests	
User	Must not notice a drain or reduction in systems performance
	The progress bars must function properly
Preconditions	Application is open
	The ability to get the system time
Success Guarantee	Aries's clock will have the same time as the system's it is running on. All Aries's threads will run correctly.

Main Success Scenarios	
1)	User opens the Aries Application
2)	All progress bars will begin to run and the clock will be at the correct time
Extensions	
a)	If the system is cannot get the time
i)	Appropriate error message displayed
ii)	User clicks ok and the system does not hang or crash
Special Requirements	Must not restrict the user's normal activities

Table 5 - System Timer Use Case

The Use Case for the systems posture guide, which allows the user to set up their workstation correctly, is detailed below in Table 6.

Correct set-up of the user's workstation	
Use Case Name	Display Posture Guide
Scope	Aries
Level	System goal
Primary Actor	User
Stakeholder & Interests	
User	The guide must be simple and easy to follow
	The guide must not contain any technical references/terms
	The guide must have a logical layout
Preconditions	The Aries application is open
	The users work means that they use a workstation
	The user has the proper equipment in order to fulfil the guidelines
Success Guarantee	A screen appears displaying a graphic of how the users workstation should be set up in order to help prevent/reduce RSI
Main Success Scenarios	
1)	The guide should appear on start-up
2)	The user can access the guide at any time but click on the "Posture Guide" button in the "File" pull-down menu
Extensions	
a)	At any time the user wants to close the guide once it is open
i)	User clicks on the close button
ii)	User clicks on the cross at the top of the window
b)	At any time the user wants to exit the application
i)	User clicks on the "Exit" button in the "File" pull down menu
ii)	User "right-clicks" on the Aries icon in the systems notification area and click "Exit" from the pop-up menu

Special Requirements	The guide must load almost instantly

Table 6 - Posture Guide Use Case

The Use Case for the system's MicroBreak is detailed below in Table 7.

Display a MicroBreak	
Use Case Name	Display MicroBreak
Scope	Aries
Level	System Goal
Primary Actor	User
Stakeholder & Interests	
User	Clear, logical layout
	MicroBreak progress bar
	No errors during running
Preconditions	Application is open
	The MircoBreak's image can be located
Success Guarantee	A MircoBreak Screen appears and stays on screen till the progress bar has completed
Main Success Scenarios	
1)	The user's work intensity reaches three or above, resulting in a MicroBreak occurring
2)	The user clicks on the MicroBreak button from the File pull down menu
Extensions	
a)	The user may choose to ignore the MicroBreak
i)	User clicks on the ignore button
ii)	User clicks on the cross at the top of the window
b)	At any time the user wants to exit the application
i)	User clicks on the "Exit" button in the "File" pull down menu
ii)	User "right-clicks" on the Aries icon in the systems notification area and click "Exit" from the pop-up menu
Special Requirements	Stays at the forefront of the screen, forcing the user to acknowledge it

Table 7 - MicroBreak Use Case

The Use Case for the system's Rest Break is detailed below in Table 8.

Display a Rest Break	
Use Case Name	Display Rest Break
Scope	Aries
Level	System Goal
Primary Actor	User
Stakeholder & Interests	
User	Clear and logical layout

	Clear and simple to understand instructions
	A graphic illustrating the stretch
	A progress bar showing the time remaining of the Rest Break
	The option to display an alternative stretch
Preconditions	Application is open
	The xml file exists where the stretch information is held
	The stretch images can be located
Success Guarantee	A screen appears displaying a graphic of a stretch for the user to perform, along with instructions on how to achieve this and the benefits of such a stretch.
Main Success Scenarios	
1)	The progress bar reaches a hundred percent. This duration this takes to reach completion can be adjusted using the strictness slider.
2)	The user clicks on the RestBreak button
3)	The user clicks on the RestBreak button from the File pull down menu
Extensions	
a)	The user may choose to ignore the RestBreak
i)	User clicks on the ignore button
ii)	User clicks on the cross at the top of the window
b)	The user may wish to see an alternative Stretch
i)	User clicks the “another stretch” button
c)	At any time the user wants to exit the application
i)	User clicks on the “Exit” button in the “File” pull down menu
ii)	User “right-clicks” on the Aries icon in the systems notification area and click “Exit” from the pop-up menu
Special Requirements	Stays at the forefront of the screen, forcing the user to acknowledge it

Table 8 - Rest Break Use Case

The Use Case for parsing and reading the XML from file is shown here in Table 9.

Reads Stretches from XML	
Use Case Name	Read XML
Scope	Aries
Level	System Goal
Primary Actor	None
Stakeholder & Interests	
User	All the stretches are read correctly from XML file and can be displayed
Preconditions	Application is open
	The XML file exists where the stretch information is held

	The XML file is complete with the correct nodes and information
Success Guarantee	All the stretches are displayed, sorted by the area of the body that they benefit.
Main Success Scenarios	
1)	All the stretches are parsed and loaded into the system successfully
Extensions	
a)	If the XML file has become damaged/corrupt
i)	The system will not crash
ii)	An appropriate error message will be displayed
b)	If the XML file cannot be located
i)	An appropriate error message will be displayed
Special Requirements	The stretches must load on start-up

Table 9 - Read XML Use Case

The Use Case for the system displaying all the stretches read from the XML file to the user is detailed below in Table 10.

Display all the Stretches	
Use Case Name	Show all Stretches
Scope	Aries
Level	System Goal
Primary Actor	User
Stakeholder & Interests	
User	The user can browse through the stretches by the area that they benefit
	The stretch instructions must be clear and concise
	The layout must be logical
Preconditions	Application is open
	The Read XML class works correctly
Success Guarantee	All the stretches are displayed, sorted by the area of the body that they benefit.
Main Success Scenarios	
1)	User clicks on the “Stretches” tab
2)	User clicks on the body area they wish to view
Extensions	
a)	At any time the user wants to exit the application
i)	User clicks on the “Exit” button in the “File” pull down menu
ii)	User “right-clicks” on the Aries icon in the systems notification area and click “Exit” from the pop-up menu
Special Requirements	The stretches must load on start-up

Table 10 - Display All Stretches Use Case

The Use Case for the system displaying the alerts for the users scheduled breaks is detailed here in Table 11.

Displays an alert for a Scheduled Break	
Use Case Name	Display Scheduled Breaks
Scope	Aries
Level	System Goal
Primary Actor	User
Stakeholder & Interests	
User	Clear and logical layout
	A title illustrating the type of break
	A graphic illustrating the type of break
	The ability to dismiss the alert
Preconditions	The Application is open
	The images need exist on the system
	The times at which the alerts are to appear have been set
Success Guarantee	A window opens displaying the type of break, when it is time for that break to be taken.
Main Success Scenarios	
1)	The user sets the time at which they want the system to alert them of their morning break, lunch break and when it is the end of the day.
2)	User achieves this my selecting the time from pull down menus
3)	The user clicks “Save Settings” button
Extensions	
a)	The user can dismiss the alert
i)	User clicks the “Dismiss” button
ii)	User clicks the cross at the top of the window
b)	If the screen cannot be shown
i)	An appropriate error message will be displayed
c)	At any time the user wants to exit the application
i)	User clicks on the “Exit” button in the “File” pull down menu
ii)	User “right-clicks” on the Aries icon in the systems notification area and click “Exit” from the pop-up menu
Special Requirements	None

Table 11 - Schedule Break Use Case

The Use Case for the system displaying the user’s activities during the session is shown here in Table 12.

Display User Actions	
Use Case Name	Display User Actions
Scope	Aries
Level	System Goal
Primary Actor	User

Stakeholder & Interests	
User	Clear and logical layout
	Whole numbers clearly expressing what a number
	A sensible scale for distance
Preconditions	The Application is open
	The monitor use class functions properly
Success Guarantee	Statistics about the user's activities are clearly shown.
Main Success Scenarios	
1)	The user opens the Aries Application
2)	Clicks on the Actions tab
3)	The users actions should be clearly shown
Extensions	
c)	At any time the user wants to exit the application
i)	User clicks on the "Exit" button in the "File" pull down menu
ii)	User "right-clicks" on the Aries icon in the systems notification area and click "Exit" from the pop-up menu
Special Requirements	None

Table 12 - Display User Actions Use Case

3.1.3 General Constraints

The main constraint for my system is that of time. If I had more time to actually focus on the design and implementation rather than the report, I believe I could achieve all the requirements and versions.

3.1.4 Assumptions and Dependencies

The system should not hang and should run on systems even with low specifications.
The system must be compatible with most operating systems, including both new and old.
The system is dependent upon Java being on the system.

3.2 Database Requirements

This system does need a database for its functionality. A later version of the system could store past days statistics in a centralised database, offering a record of activities that could be analysed to help prevent bad habits.

3.3 Non Functional Requirements

3.3.1 User Interfaces

The external users of this system are the people operating the computer on which it is installed. They will be presented with the main home screen of the system when the program is launched. This system is designed for just one class of user.

3.3.2 Hardware Interfaces

This can be any form of computer terminal, be it a desktop or a laptop.

3.3.3 Software Interfaces

The operating system can be any version of Windows, Linux/Unix or Mac that supports Java.

3.3.4 Design Constraints

No specific constraints exist.

3.3.5 Quality Assurance

The system must not “hang” or crash during its use. The system is also expected not to cause any slowing down of the system and must not be resource hungry.

3.3.6 Memory Requirements

Memory requirements should not exceed that of a standard computer manufactured in 2006 onwards. This typically is of 512MB RAM and more. This is to ensure that the system will be able to run on as wide a consumer base as possible.

3.3.7 Packaging Requirements

The software will be submitted as a zip file consisting of all the source code and precompiled binaries. An auto generated installation script may also be provided depending on time constraints.

3.3.8 Legal Requirements

The software must comply with the Data Protection Act of 1998 in order to protect the user’s privacy. Care must be taken to adhere to the copyright restrictions on third party components used throughout the development of my system.

3.3.9 Regulatory Requirements

Throughout the design and implementation as well as when the completed system is running, the uppermost care must be taken in order to adhere to the University’s guidelines regarding to cheating and plagiarism.

3.4 Software System Attributes

3.4.1 Reliability

My program must be both reliable and robust in order for it to be a successful piece of software that can be taken on board by both companies and individuals.

3.4.2 Availability

My system will be widely available for both commercial and personal use.

3.4.3 Security

It is of vital importance that my system is secure and is not at risk of an attack that could jeopardise the user.

3.4.4 Maintainability

Throughout the making of my system, adherence to software engineering protocols will be strict, with sufficient comments and modularity included throughout.

Each class must contain the standardised project header containing the author information, version number and date, as well as the history of any changes made.

A separate manual giving detailed instructions on how to maintain the system and a detailed overview of the system will be provided once the implementation phase has been fulfilled.

3.4.5 Portability

My system should be able to be installed on various Operating systems such as Windows, Linux and Macintosh in order to cater for all potential users.

3.4.6 Performance

My system must not lag or hang through use. This is of vital importance in order to offer a stress free experience for the user of my system.

3.4.7 Accessibility

The user interface should be accessible to those with disabilities and where appropriate adhere to relevant guidelines.

3.4.8 Learnability

The user should be able to learn how to use the system without tuition. They will be aided with any problems by the system's help screen.

3.4.9 Error Handling

The system should be fault tolerant with correct error handling at all times.
In general, unrecoverable errors should not occur.

Chapter 4
Design and Implementation Decisions

This chapter discusses and goes into more detail about how the functionality in the last chapter is going to be realised into a physical system.

4.1 Application Type

There are various types of application that this proposed system could take. Although from the original concept, the system was always going to be a traditional desktop application. I took it upon myself to research the topic and looked into the other possibilities that were out there and too see if they provided a more elegant alternative.

The alternative that was looked into was that of having the system deployed as a web based application. Such a system would mean that the only requirements that the user would need would be that of a web browser with the appropriate plug-ins. However, even with the vast improvement and advancement in web based technologies, it soon became clear that the functional requirements that are necessary to my system could not all be fulfilled.

This fact amalgamated with nearly all my programming and development experience being with that of the traditional desktop application type further cemented the conclusion of going on the traditional path.

4.2 Development Language

As a non-functional requirement, my system was implemented with the ambition of achieving full platform independence. This was a key aspect when it came to choosing the programming language that would be used to develop the system. Another important aspect to consider when choosing the correct language was to factor in how much experience I had had using the language and how much time would be needed to learn new techniques and syntax.

The language that was decided upon had to be in widespread use and have ample documentation available so that in the future the system could be developed further by other developers.

The following Object Oriented programming languages where considered and the evaluation by comparing and contrasting these languages is shown in Table 13.

Language	Platform	Owning body	Open source	Support	Experience	Notes
C#	Windows Only	Microsoft	No	Available	None	There is support available through the open source “mono” project. Not all C#

						features are fully documented here however.
C++	Platform Independent	Many Implementations	Varies	Widely used with substantial documentation available	Limited	For every platform the source code needs to be recompiled. Different implementations support different features.
Java	Platform Independent	Sun Microsystems (now owned by Oracle)	Yes	Current with vast amounts of documentation available	Extensive	Wide support on any platform that is able to run a Java Virtual Machine (JVM). Apple Mac support is typically six months behind that of current release.

Table 13 - Comparison of Object Oriented Programming Languages

As Java fulfils all the system requirements, combined with being the language I have must had experience with, Java is the language I have chosen to develop my project in. However, I later discovered that not all of the proposed requirements could be fulfilled using purely Java so I had to integrate some C code into my predominantly Java project.

4.3 Architecture

The ability of the system to be both appealing and fast coupled with the fact that the system needed to be reliable and robust meant that the choice of system architecture was an important decision to get right.

For details of the research carried out on the possible architectures available see Appendix E.

4.3.1 Chosen Architecture

As the sole developer of the system, the architecture pattern that I set out to accomplish falls under that of application architecture. Layered architecture, which is a decomposition of services where most interactions happen only between neighbouring layers, is the pattern under application architecture that best fits my proposed project. When developing the system emphasis is put on the function of individual elements and how they affect the design patterns. Emphasis is also put on software; this is to see if the desired goal is achievable. As I am the only member of the team, less emphasis is put on the "team" as I know what need to be achieved and the standard of work that is to be achieved.

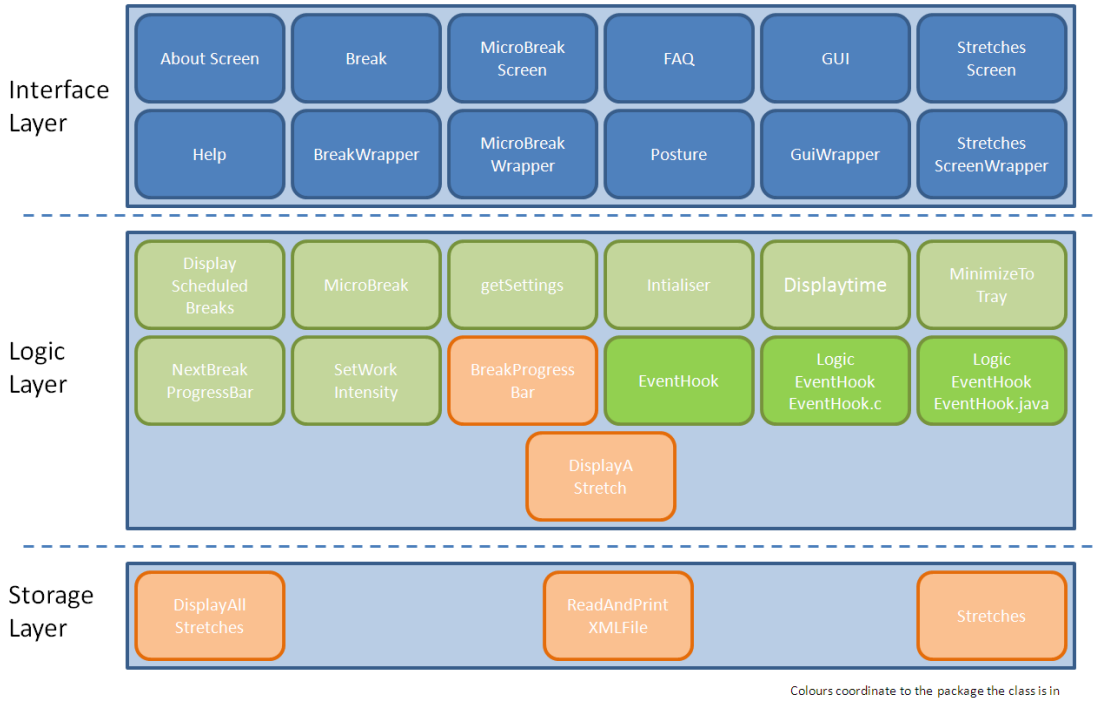
4.3.2 Storage

Although the system at present does not have a database, it could easily be modified in the future to incorporate one into its architecture. If the system was to store statistics about the user's activities in future versions of the system a database would be needed in order to hold these statistics. The need for a database would also need to be realised if the system was to incorporate password protected settings. These passwords would need to be hashed into a database for security.

The decision to hold the information for the stretches in XML format rather than to store them in a central database was one made quite early on. The use of an XML file was believed to be a more elegant way of storing the information as it meant that portability would be greatly increased and the need of dependencies would be reduced - as the XML file could be bundled with the Aries package unlike a database.

4.3.3 System Components Interaction

The overall layout of the components in the system is that of the classic three-tier architecture familiar in software engineering. Although my system does not have a database, that is so often associated with this form of layout, it does however request and receive data from an XML file and has a logic and an interface layer. This is shown in figure 10 below.



Key: Blue = GUI package, Green = Logic package, Orange = Stretches package

Figure 10 - Three Tier Architecture of System

The classes in the interface layer handle all the systems GUI's and features that are presented to the user. The logic layer deals with all the calculations and the holds

the main workings of the system, where as the storage layer handles the input of data from external sources, such as the XML file.

4.4 Security

As previously discussed in section 4.3.2 user passwords would need to be hashed using the SHA-1 algorithm if they were going to be stored in a central database.

Hashing a password is a technical term for disguising it with the help of an algorithm which in this case is the SHA-1 algorithm. The process of hashing is a one way process and it should be impossible to recover the original text once it has been hashed. Therefore the process, results in it being impossible for database administrators or potential hackers to retrieve passwords by viewing the database tables.

It is vital that my system be secure due to its nature of recording the users key-stokes. The system will be extensively tested and audited before it is made commercially available to ensure that malicious code cannot gain access to personal data.

To ensure that the system cannot be used like a 'Keylogger' virus it was decided early on in proceedings that the system would not record individual key properties.

A Keylogger works by recording all the keys pressed by the user during a session, and as a result private data such as passwords and credit card numbers can be retrieved and sent to the intruder for him to exploit at will. Likewise the program may look to steal product keys, allowing the intruder to steal accounts and virtual data.

4.5 System Protocols

The goal for the first prototype of the system was to keep it as simple as possible and have just the core requirements implemented, functioning and tested.
The first version of the system should support the following functionality:

4.5.1 Prototype 1.0

Details for prototype 1.0 are show below in Table 14.

Listeners
System shall be able to tell the user how many keys that have been pressed during the present session
System shall be able to tell the user how far the cursor have travelled in centimetres
Stretches
System shall be able to allow the user to see stretches at any time These stretches will cover areas such as Legs, Back, Neck, Hands and Eyes.
Breaks

User shall be able to set the duration between rest breaks (2,6,10 minutes)
The system will show breaks according to these settings
Basic Functionality
User shall be able to exit the system at anytime
User shall be able to see information about how to use the system at anytime. This will be in the form of a help screen.
The user shall be able to see information about the system. An About screen for displaying why the program was made.
The system should show the Posture Guide on start-up

Table 14 - Prototype 1.0

4.5.2 Prototype 1.1

The aim of the next prototype is to build in more functionality around the bare bones of the system. The proposed functionality improvements are detailed below in Table 15.

Threads
System shall be able to show the user how long it is till the next break
System shall be able to show the overall time (in hrs:mins) the system has been in use (Total time the user is on the computer)
The user should be able to set a time for when the system will alert them of when it is time for the morning break
The user should be able to set a time for when the system will alert them of when it is time for the lunch
The user should be able to set a time for when the system will alert them of when it is the end of the day
Breaks
User shall be able to ignore a break if non-compulsory

Table 15 - Prototype 1.1

4.5.3 Prototype 1.2

The next prototype is again building on the foundations laid by the previous two versions. Details of the new features are listed below in Table 16:

Functionality
System shall be able to tell the user how intensely they are working: This will be calculated by how many keys have been pressed during a time period, such as 5 minutes Intensity will be measured on a scale of 5.
Breaks
User shall be able to set the duration of uninterrupted work
System shall be able to allow the user choose where they are allowed to ignore breaks or not

Table 16 - Prototype 1.2

4.5.4 Prototype 2.0

This prototype details the foundations of the next version of the system. Prototype 1 will be the level of acceptable functionality that would mean that the system could be put into the market. Prototype 2 aims to build on version 1 by implementing the following functionality (shown in Table 17).

Functionality
System shall be able to Show the user what programs they have been using from the present session
System shall be able to determine when the next micro break shall be depending upon intensity level.
System shall be able to tell the user how many keys their left hand has pressed
System shall be able to tell the user how many keys their right hand has pressed
User shall be able to have the system blacken the background during breaks
User shall be able to disable the mouse during breaks

Table 17 - Prototype 2.0

4.5.5 Prototype 2.1

This prototype builds on that of 2.0 and implements additional features for the system. These features are detailed below in Table 18:

Functionality
The system should have a pre-break sound
The system should have a post-break sound
The system should have a pre-break visual
The system should allow the user to set the frequency of breaks for the user by selecting from the choices of: RSI prevention, Got RSI or Bad RSI
Posture guide
If unselected, system should not show posture guide on start-up

Table 18 - Prototype 2.1

4.5.6 Prototype 3.0

Prototype 3 is the next version of the software. It builds on version 1 and 2 by introducing security aspects and extra improvements to the overall performance of the system. Detailed here in Table 19.

Functionality
The system should give the user the option to show the correct posture guide on start-up
The system should show posture guide on start-up (unless unselected)
Functionality and Security
The system should have a password protected settings

Table 19 - Prototype 3.0

4.6 UML Sequence Diagram

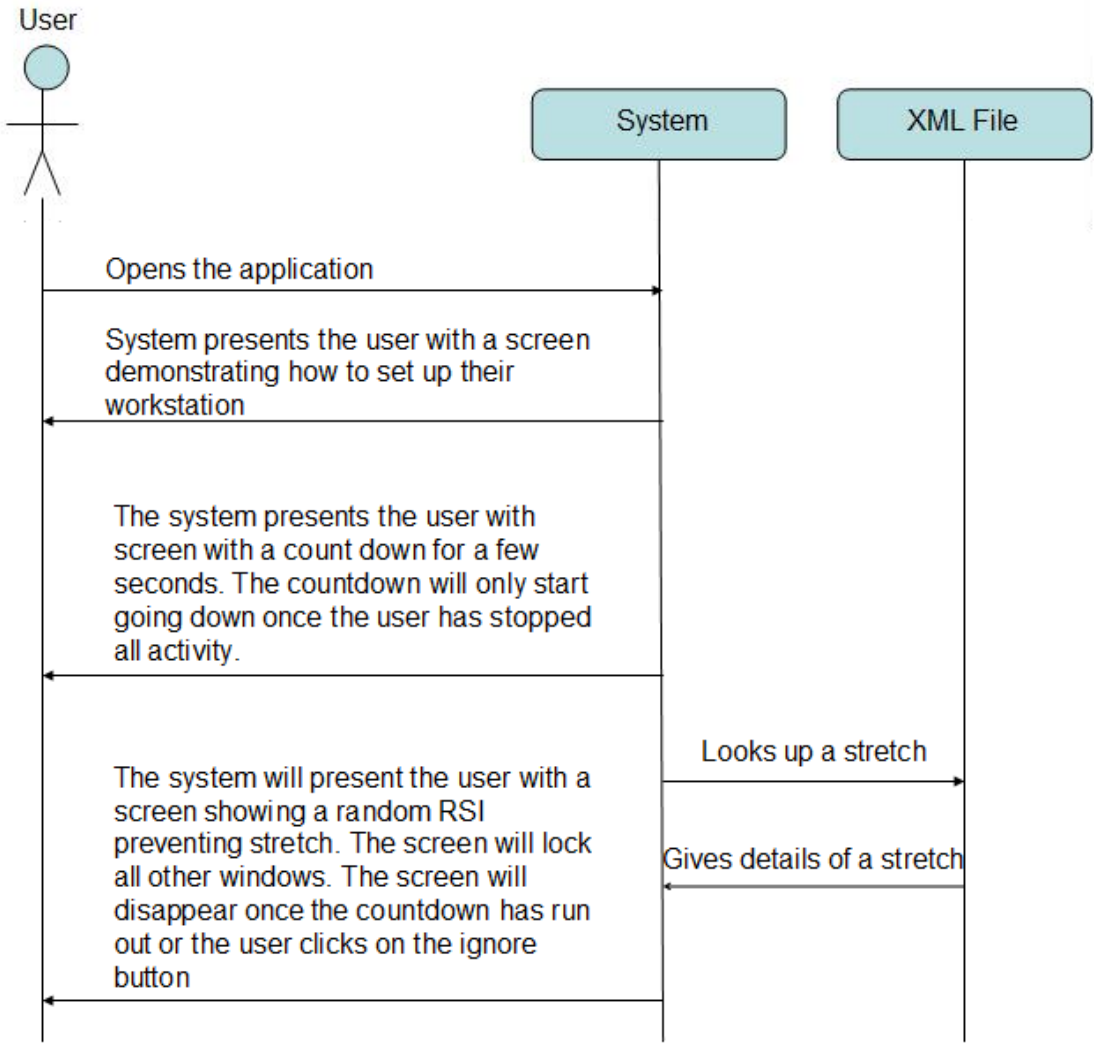

User
System
XML File
Opens the application
System presents the user with a screen demonstrating how to set up their workstation
The system presents the user with screen with a count down for a few seconds. The countdown will only start going down once the user has stopped all activity.
Looks up a stretch
The system will present the user with a screen showing a random RSI preventing stretch. The screen will lock all other windows. The screen will disappear once the countdown has run out or the user clicks on the ignore button
Gives details of a stretch

Figure 11- UML Sequence Diagram illustrating the basic interactions between the user and the system

4.7 User Interface Design

When initially designing the Graphical User Interface (GUI) research into existing systems offering similar services was carried. It soon became clear on researching systems that were similar to mine that they all stuck to a similar GUI design and it was decided that my system would follow this trend. It was thought that such a decision would help increase user acceptance as usability is a vital aspect of a product such as this one where its whole objective is to decrease stress and RSI.

The main GUI was designed to take the best aspects from existing systems. The systems used for this process were RSI Warrior (Figure 12) and Workpace (Figure 13).

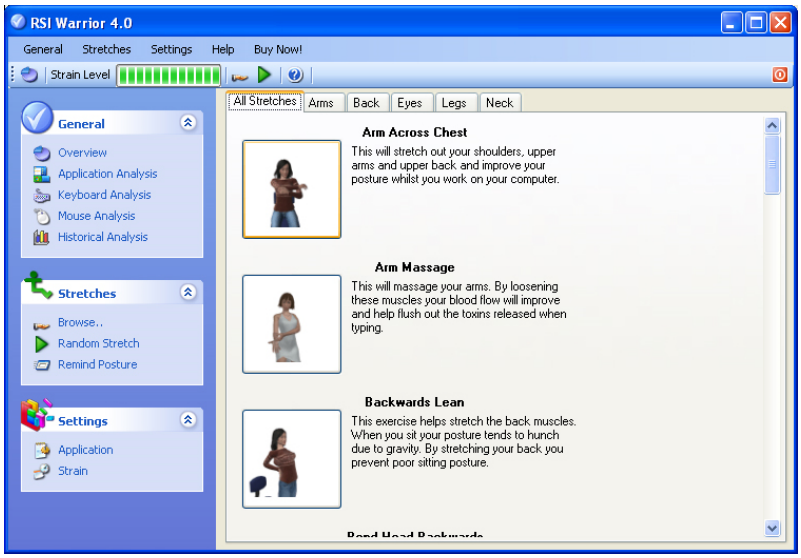


Figure 12 - RSI Warrior's GUI

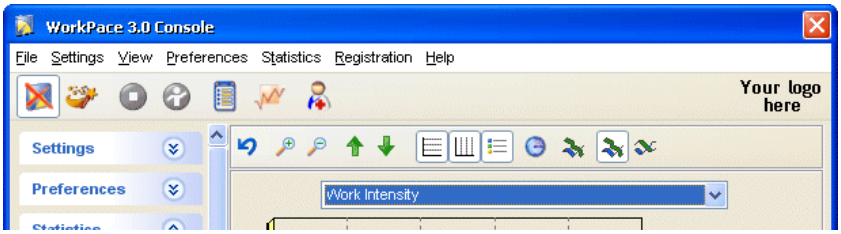


Figure 13 - Workpace menus and GUI

Both these GUI's consist of user friendly icons for buttons and have easily accessible pull down menus on the side. In the middle right of the interface is the main focus, where the users desired content is displayed. Aries design follows in a similar vein to these systems and has the main functionality displayed along the top rather than down the right hand side (shown below in Figure 14). The GUI also consists of two pull down menus offering further features, such as the help guide.

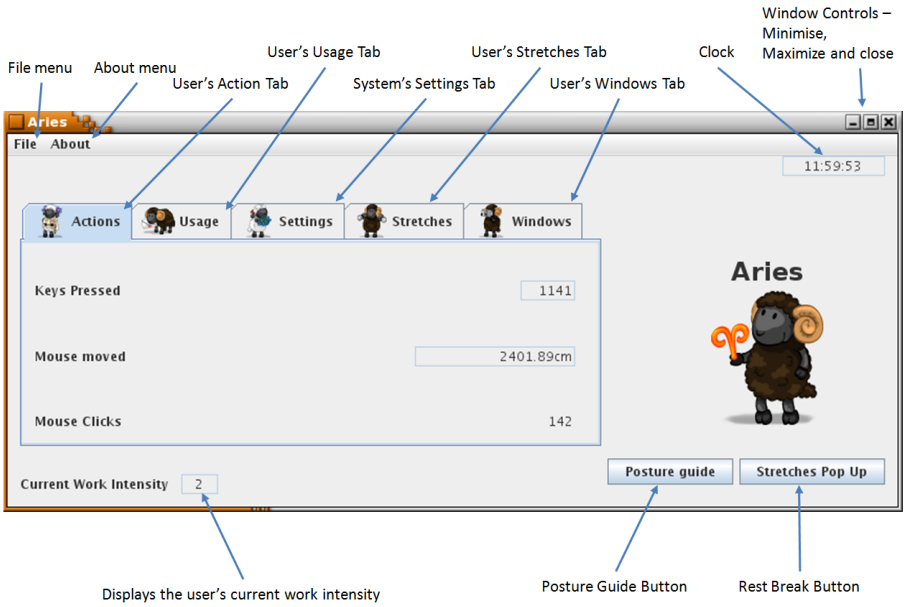


Figure 14- Aries Main GUI

4.7.1 Stretches GUI

Research similar to that of the main GUI was carried out in order to discover the best layout for the breaks to consist of. There are many existing systems that offer such breaks. Some of these are detailed below.

4.7.1.1 Rest Break

RSI Warrior offers as part of its system a stretch player. This consists of an animated woman performing the stretches, written instructions and spoken instructions. This results in the user knowing exactly what it is they need to fulfil in order to be doing the stretch properly. A screen shot of the stretch player can be found in Figure 15 below.

RSI Warrior 3.0
Stretch Player
Instructions
Stand with your feet shoulder width apart and hands on lower back. Looking straight ahead, slowly lean back, keeping legs straight.
Press play to start stretch

Figure 15 - RSI Warrior Stretch Player

Workpace is another system that offers RSI preventing stretches. Similar to my system Workpace presents the user with a graphic of a stretch and text describing the aim of the stretch as well as its starting position and method. Workpace also allows the user to pause a stretch or to cancel the stretch altogether. Workpace's

breaks are comprised of four stretches and have a countdown timer for each stretch visible on the screen. An example of such a stretch can be seen in Figure 16 below.

Arm and Shoulder Stretch
Aim
To stretch your hands and forearms and to encourage movement of your shoulder.
Starting Position
Sitting up tall, interlink your fingers, palms facing towards you. Roll your palms down so they face away from you (stage 1).
Method
Gently straighten your elbows. Hold for 2 seconds. Now take your arms above your head. Feel a stretch in your arms, shoulders or upper back (stage 2). Hold for 5 seconds then bring your arms down.
Repetitions: 3
Cancel
Pause
1/4
30

Figure 16 - Workpace Break

It was decided to follow a similar design trend to that of Workpace's for Aries because of its simple, logical layout which portrays the intended message with ease. The system's Rest Break would display a graphic of a stretch combined with text describing what the stretch is designed to achieve. The pop-up screen will also display the title of the stretch and have a countdown (progress bar) that would show the user the time remaining until the break disappears. The screen will also have a 'close' button. When this is pressed the screen will disappear. Aries' Rest break screen design can be seen below in figure 17.

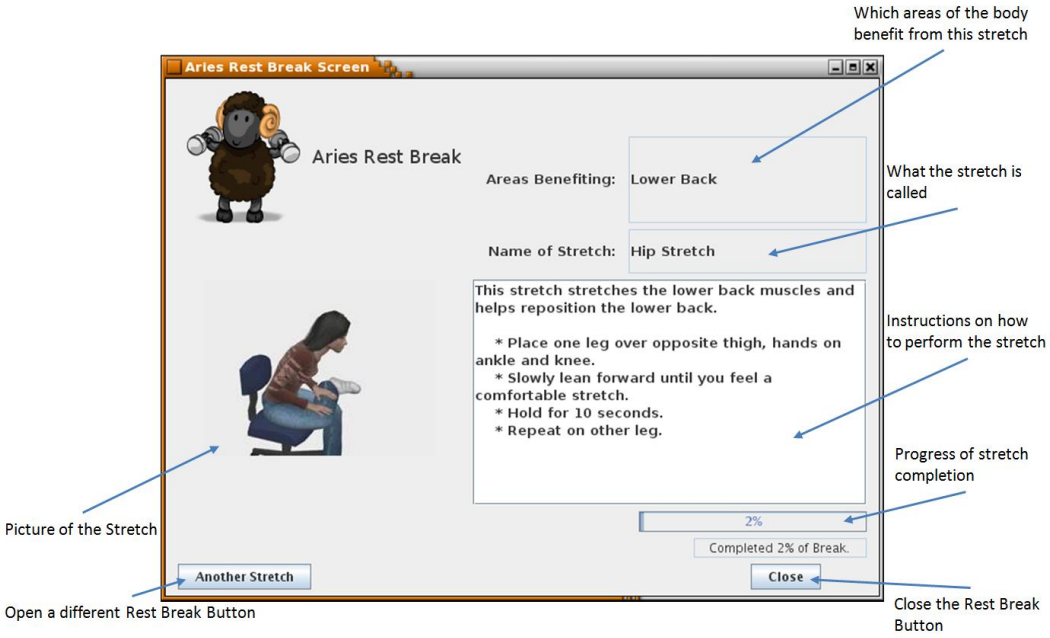

Which areas of the body benefit from this stretch
Aries Rest Break Screen
Aries Rest Break
Areas Benefiting:
Lower Back
What the stretch is called
Name of Stretch:
Hip Stretch
This stretch stretches the lower back muscles and helps reposition the lower back.
* Place one leg over opposite thigh, hands on ankle and knee.
* Slowly lean forward until you feel a comfortable stretch.
* Hold for 10 seconds.
* Repeat on other leg.
Instructions on how to perform the stretch
Progress of stretch completion
2%
Completed 2% of Break.
Picture of the Stretch
Another Stretch
Close
Open a different Rest Break Button
Close the Rest Break Button

Figure 17 - Aries Rest Break Window

4.7.1.2 MicroBreak Screen

Upon designing the appearance of the MicroBreak screen, research was carried out on various systems that offered such a function. Many consisted of a similar interface of a simple pop-up screen comprising of a countdown timer (progress bar) and a cancel button.

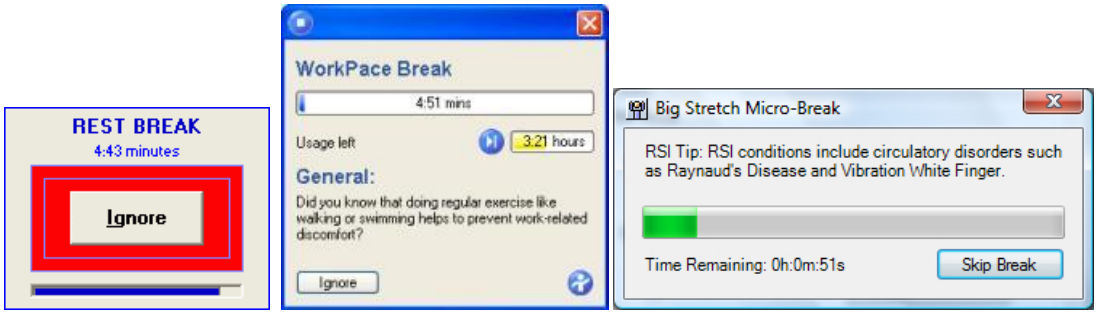


Figure 18 - Break Reminder, Workpace and Big Stretch

This appeared to be the best solution to the design of such a screen due to its simplicity. Aries's MicroBreak follows the same design principles but with a view of being more user friendly and aesthetically pleasing. Aries MicroBreak design can be seen below in figure 19.

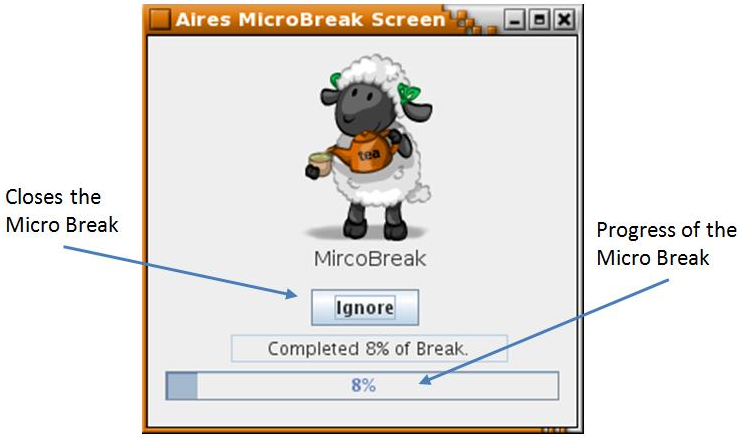


Figure 19 - Aries MicroBreak

4.7.1.3 Schedule Break Screen

The Aries system also has the functionality to display the user with alerts throughout the day at pre-determined times. This allows the user to set times for when they want to have their morning break, lunch and when it is time for them to go home. These screens where designed to be user friendly and consist of a picture and text both describing the type of break. This allows the user to understand instantly what the break alert is. The Aries design for the three breaks can be seen below in figure 20.

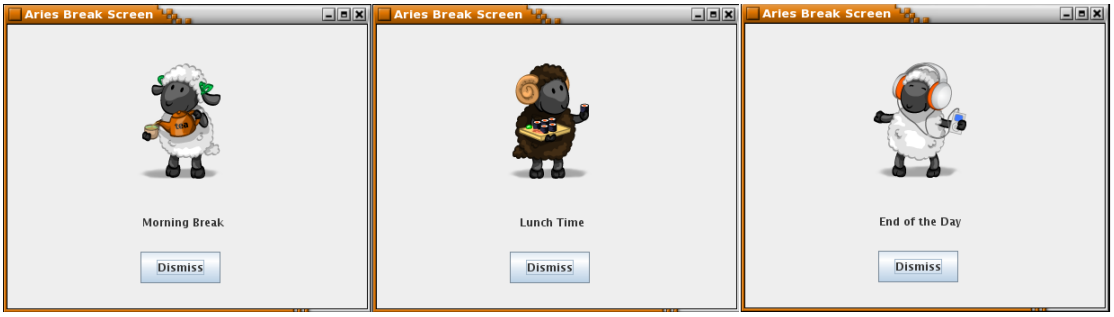


Figure 20 - Aries Scheduled Break Screens

4.7.2 About Screen

The design principles behind the About screen were to produce a simple screen that could convey the idea and drive being such a system as Aries. The design was to have a simple screen displaying the systems logo and a short text description of why it was developed. Aries's About screen can be seen in figure 21.

Aries About Screen
RSI Prevent
RSI Prevent is Software designed and developed to help reduce RSI in the workplace.
This system was designed and implemented by Peter Kindness In the Java langauge, as part of his 4th Year Computing Honours Project at Aberdeen University.
Close

Figure 21 - Aries About screen

4.7.3 Help Screen

The design of the Help screen had to be as simple as possible due to its very nature. It is not beneficial having a confusing help screen which the user may need more help to navigate. Therefore the design of the Help screen meant that it would consider of a pop-up screen displaying tabbed panes. Each pane would cover an area of the system. The result of this is that the user will be able to easily navigate the help system with both ease and speed. The pane will then have information about how to use that part of the system depending on which tab it is that has been selected. Aries's help screen design can be seen below in figure 22.

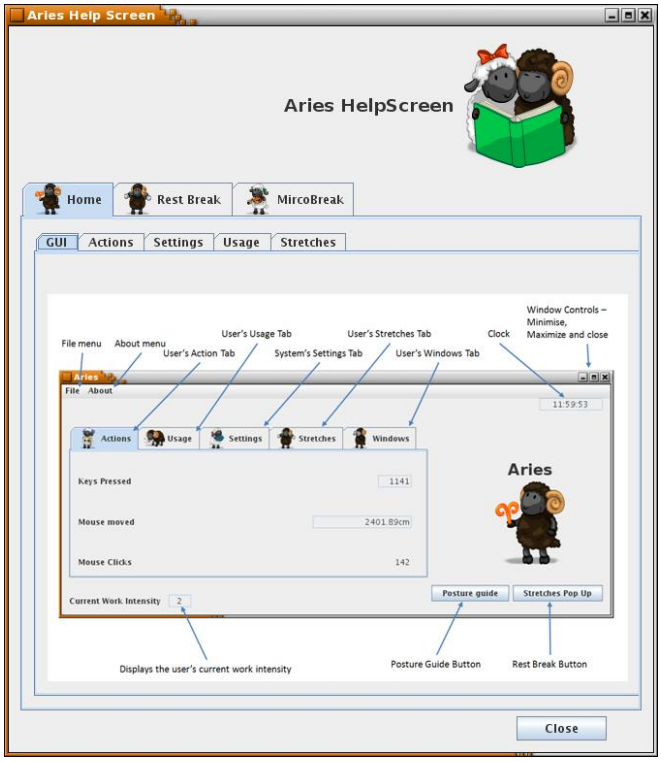


Figure 22 - Aries Help Screen

4.7.4 Posture Guide

As the posture guide may be shown at start-up, it is essential for the screen to be very easy to understand as it may be the first contact that a user has had with the system. It was decided that it would be far more effective to have a diagram showing the user how to set up their workstation rather than a detailed text description that may tempt the user to skip the step altogether. This is shown in figure 23 below.

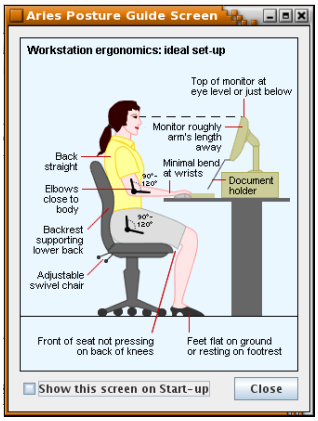

Aries Posture Guide Screen
Workstation ergonomics: ideal set-up
Top of monitor at eye level or just below
Monitor roughly arm's length away
Minimal bend at wrists
Back straight
Elbows close to body
Backrest supporting lower back
Adjustable swivel chair
Document holder
Front of seat not pressing on back of knees
Feet flat on ground or resting on footrest
Show this screen on Start-up
Close

Figure 23 - Aries Posture Guide

4.7.5 Accessibility

Accessibility is an important issue to consider whilst designing the system as it is being targeted at as wide a consumer base as possible. The addition of sound in the future would be an important accessibility increasing feature. The system is also being designed with this in mind, to prevent any accidental alienation of consumers that could lead to legal issues.

4.7.6 Dimensions

The default dimensions of the GUI are another important feature to consider with monitors coming in all shapes and sizes. With the increase of netbooks and with laptops getting increasingly smaller as technology improves it is an issue worth addressing. The system has been set with the default dimensions which are shown in Table 20. All the windows will appear in the middle of the users screen and all are resizable; this is an attempt to allow the user to customise the screens to tailor to his needs.

Window	Width	Height
About	400	349
Scheduled Break	400	315
RSI – FAQ	683	431
GUI	811	366
Help	665	750
Micro Break	270	279
Posture Guide	324	417
Splash	400	381
Rest Break	701	514

Table 20 - Window Dimensions

4.7.7 Navigation

The navigation throughout the system has been designed to be consistent in order to aid usability and make the ease at which a user becomes familiar with the system much greater. The global navigation of the system is constantly at the top of the GUI. This is now standard design in many systems and thus should be instantly recognisable to the user. Local navigation varies between panes and screens but has been designed to be as simple and easy to use as possible.

4.8 Proposed Database Design

Although in the scope of this project the database is a relatively minor feature due to the fact that it is only needed to implement a future function, it has been decided to layout the structure here in order to aid future development. The database would need to hold the users details and their passwords, which in turn would need to hashed to aid security.

Research was carried out to explore the various databases available to be used in the system. Both integrated databases such as JavaDB and independent databases, such as Derby were explored, however MySQL was chosen for the following reasons:

- Open Source and free
- Accessible from Java
- Extensive front-end tools, such as phpMyAdmin
- Platform Independent
- Run and supported by many hosting companies
- Extensive support and documentation available

4.8.1 Database Schema

```
rsiPrevent_users {
        rsip_loginname      varchar     (32)
        rsip_password       varchar     (256)
        rsip_admin    char  (5)
}

rsiPrevent_settings{
        rsip_ignore   char  (5)
        rsip_strictness     char  (5)
        rsip_monitorWindows       char  (5)
}
```

4.8.1 EER Diagram

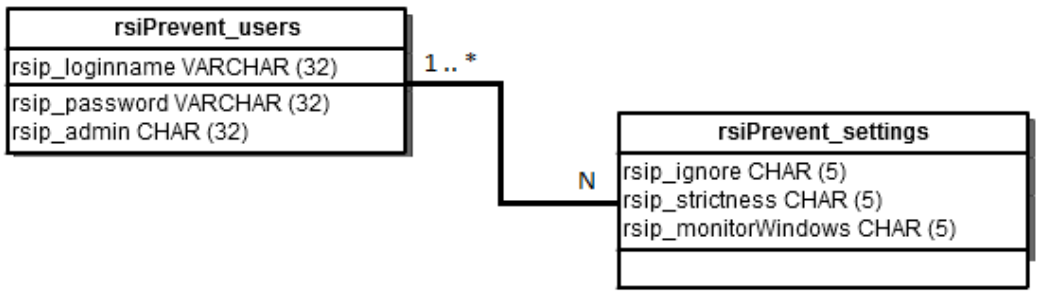


Figure 24 - Database EER diagram

Chapter 5
Development and Implementation of Core Components

This chapter discusses in more detail how the functionalities previously mentioned in the last chapter were implemented, as well as outlining any deviations in the final product from that of the original design, due to problems faced in the development process.

5.1 Keystroke logging

Keystroke logging is when a program tracks the keys struck on a keyboard, most commonly in a covert manner so that the user is unaware that their actions are being monitored. Due to the very nature of such a program very limited documentation is available. Even though there is a lack of documentation, many malicious codes exist of this nature, reassuring me that creating such code is possible. However, due to security concerns, an early decision was made not to record any information on the keys pressed, just that a key was pressed. Implementing this for my system turned out to be a much more difficult than first thought.

In order for this to be achieved, I first needed to create an operating system hook, which would listen to the OS and respond accordingly when a key was pressed. This however needed to be done using C++, which is where the majority of my difficulties occurred. Although my knowledge of C++ is limited, after much research I managed to write code that should in theory work. However various libraries were needed in order for this to work and the following error message was appearing due to the fact that the University's C libraries were very old and outdated.

Error message:

```
Exception in thread "main" java.lang.UnsatisfiedLinkError:
 /home/student/cs/cs02pk/NetBeansProjects/KeyListener/lib/libJXGrabKey.so:
/usr/lib/libstdc++.so.6: version `GLIBCXX_3.4.11' not found (required by
/home/student/cs/cs02pk/NetBeansProjects/KeyListener/lib/libJXGrabKey.so)
```

After several meetings with the university's technician and after updating the systems C++ compilers still no solution was present. This was mainly due to the fact that the library (JXGrabKey) that this code needed was relatively newly released and as a result was not fully compatible with all the Linux environments. This meant that a rethink was needed and several possible solutions presented themselves.

The first problem that presented itself was a simple privileges problem. Due to the fact that the system was being implemented at University, this meant that I as the developer had limited system privileges. This meant that I could not install any additional programs onto my system or gain access to the kernel that is needed to listen to user events. The event listener would need to sit between the kernel and

the security level and listen to the user's actions. A possible solution to this was to get the technician to install Qemu on my system. Qemu would allow me to run a virtual machine on my system, which I could have full control of. This again however was a rather impractical solution to the problem as it would require both the technician to install it and an additional twenty gigabytes of storage for my system, as Qemu is effectively a big file on the system.

After further research it was discovered that writing a system hook and key listener on the windows environment is much easier than writing it for the Linux or Macintosh environment. This is due to the low level security protocols of the differing systems, with both Mac and Linux being based on the UNIX kernel. A specific search on the internet will present several possible implementations for keylisteners on Windows, but after extensive searching no support could be found for the Linux environment. However due to the fact that all the systems in the University's computing department are currently running Centos, a Linux environment, this posed a problem.

However eventually, a solution was found. An eventlistener was successfully created in C that would work on the University systems and any other Linux system. This eventlistener works by using a system hook and listens to the events travelling between the kernel and the security level. The listener returns the events to the user as the user can not access the kernel. This only works for the user that the system is logged in as though for security reasons. Events for another user would be encrypted and so the listener would not work. A snippet of the C code can be seen below in figure 25. The eventlistener refreshes every 15 milliseconds and is relatively efficient, taking up only a few per cent of the computers CPU power.

```
16 JNIEXPORT jlong JNICALL Java_Logic_EventHook_EventHook_getMouseDistance(JNIEnv *env, jobject jobj)    { return distance; }
17 JNIEXPORT jint JNICALL Java_Logic_EventHook_EventHook_getMouseClickCounter(JNIEnv *env, jobject jobj)    { return counterMouseClicks; }
18 JNIEXPORT jint JNICALL Java_Logic_EventHook_EventHook_getKeyPressedCounter(JNIEnv *env, jobject jobj)    { return counterKeyPressed; }
19 JNIEXPORT jint JNICALL Java_Logic_EventHook_EventHook_getMouseX(JNIEnv *env, jobject jobj)        { return rx; }
20 JNIEXPORT jint JNICALL Java_Logic_EventHook_EventHook_getMouseY(JNIEnv *env, jobject jobj)        { return ry; }
21
22 JNIEXPORT void JNICALL Java_Logic_EventHook_EventHook_reset(JNIEnv *env, jobject jobj){
23     counterMouseClicks = counterKeyPressed = distance = 0;
24 }
25
26 JNIEXPORT jint JNICALL Java_Logic_EventHook_EventHook_initHook(JNIEnv *env, jobject jobj){
27     pthread_t thread;
28     display=XOpenDisplay(0);
29     root = XRootWindow(display,0);
30     XSelectInput(display,root,KeyReleaseMask);
31     return pthread_create(&thread, NULL, hookLoop, NULL) ;
32 }
33
34 int inline idist(int x1, int y1, int x2, int y2){
35     if ((x2 -= x1) < 0) x2 = -x2;
36     if ((y2 -= y1) < 0) y2 = -y2;
37     return (x2 + y2 - (((x2>y2) ? y2 : x2) >> 1) );
38 }
39
40 void* hookLoop(void *arg){
41     bool clicked = false;
42     char pastKeys[32] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}, keys[32];
43     unsigned int modifs;
44     int wx,wy,oldx,oldy,mask=Button1Mask|Button2Mask|Button3Mask|Button4Mask|Button5Mask;
45     XQueryPointer(display,root,&root2,&windowIn,&oldx,&oldy,&wx,&wy,&modifs);
46     for(;;){
47         XQueryPointer(display,root,&root2,&windowIn,&rx,&ry,&wx,&wy,&modifs);
48         if(!clicked && (modifs&mask)>0) counterMouseClicks++, clicked = true;
49         if(oldx != rx || oldy != ry) distance += idist(oldx, oldy, rx, ry);
50         oldx = rx; oldy = ry;
51         if((modifs&mask) == 0 && clicked) clicked = false;
52
53         XQueryKeymap(display, keys);
54         for(int i = 0; i<32; ++i) if(keys[i] !=0 && keys[i] != pastKeys[i]) { counterKeyPressed ++; break; }
55         for(int i = 0; i<32; ++i) pastKeys[i] = keys[i];
56
57         usleep(50);
58     }
59 }
```

Figure 25 - Event Listener Hook written in C

With the eventlistener working and written in C, the next task was to integrate the C code into Java and my project.

5.2 Integrating C with Java

In order to achieve the desired listeners that are needed for my system to be able to successfully listen to the users actions, various sources needed to be written and compiled in the programming language C. This C code then needed to be integrated into my predominantly Java written project. This turned out to be rather more difficult than was first expected. This section explores the process involved, difficulties encountered and the solutions to these difficulties.

5.2.1 Define a Native Method

The following Java source code is a simple example of what is needed in the Java class. This particular method has one method and a static code segment:

```
class CppNumberGetter {
public native int getNumber();

static {
System.loadLibrary("number");
}
}
```

The implementation for the CppNumberGetter class's getNumber() method is written in another programming language and is acknowledged by the use of the native keyword in the method's definition.

The implementation for getNumber() is supplied in a separate native language source file. Native methods, like other methods must be defined within a Java class.

5.2.2 Load the Library

The native language code that implements the above method must first be compiled into a shared library and then loaded into the Java class that requires it. I achieved this by using the following Linux command in the terminal:

With csh:

```
setenv LD_LIBRARY_PATH "path"
```

With bash:

```
export LD_LIBRARY_PATH="path"
```

The implementation of the naive method is mapped to the java class's definition once the library has been loaded.

The static code block below, taken from the CppNumberGetter class loads the correct library named “number”. A class's static code block is executed by the runtime system when it loads the class.

```
static {
        System.loadLibrary("number");
}
```

5.2.3 Create the Main Program

Within a separate source file, in this case called Main.java, a Java application that instantiates CppNumberGetter and calls the getNumber() native method needs to be created.

```
class Main {
       public static void main(String[] args) {
              CppNumberGetter jniObject = new CppNumberGetter();
              System.out.println("Cpp told me " + jniObject.getNumber());
       }
}
```

As show above a native method is called in much the same way as a regular method might be called. After the Java code was compiled, using the javac utility program, and as a result the .class files have been produced, an .h file was needed to be created. This .h file (header file) was generated using the CppNumberGetter java class using the javah utility program. This header file defines the structure that represents the CppNumberGetter class on the native language side, and provides a function definition for the implementation of the native method getNumber() defined in that class. The javah places the new .h file in the same directory as the .class files and takes its name from the Java class, so in this case CppNumberGetter.h.

```
#includejava example/CppNumberGetter.h
```

The header contains a struct definition for the structure named class CppNumberGetter. The members of this structure parallel the members of the corresponding Java class. The members of the struct can be used to reference class instance variables from the native language functions.
The header also contains the line:

```
HandleTo(CppNumberGetter);
```

This line creates a typedef struct named HCppNumberGetter, which is used to pass objects between Java and another language (in this case, C).
In addition to the structure that mimics the Java class, there is also a function signature that looks like this:

```
extern "C" {
JNIEXPORT jint JNICALL Java_CppNumberGetter_getNumber
  (JNIEnv *, jobject);
```

```
}
```

This is the definition to the function that provides the implementation for the CppNumberGetter class's native method getNumber().

5.2.4 Create a stubs file

In addition to the .h file that was generated, javah -stubs was used to generate a stubs file. The stubs file contains native language code that provides the “glue” that holds the Java class and its parallel native language structure together.

5.2.5 Writing the Native Method Implementation

After all this had been achieved the C code can be written.
The C function must have the same function signature as the one generated with javah into the CppNumberGetter.h file. The function signature generated for the CppNumberGetter class's Java_CppNumberGetter_getNumber() native method looks like this:

```
extern "C" {
JNIEXPORT jint JNICALL Java_CppNumberGetter_getNumber
  (JNIEnv *, jobject);
}
```

The implementation for this is a simple function that will return the integer 4.

5.2.6 Creating a shared library

Using the stubs file and the native method a shared library needed to be created. This was achieved by compiling the two files into a library in the terminal using the following command:

5.2.6 Running the program

The program should now run and display the result “Cpp told me 2”.

5.2.7 Problems/difficulties

A problem occurred when trying to load the shared library from inside Netbeans, but after further research a solution was found. The following lines of code is that solution.

```
private static void loadLib() {
        try {
                // have to use a stream
                InputStream in =
EventHook.class.getResourceAsStream("/Logic/EventHook/libhook.so");
                // always write to different location
                File fileOut = new File(System.getProperty("java.io.tmpdir")
+ "/" + "/Logic/EventHook/libhook.so");
```

```
                java.io.OutputStream out =
FileUtils.openOutputStream(fileOut);
                IOUtils.copy(in, out);
                in.close();
                out.close();
                System.load(fileOut.toString());
        } catch (Exception e) {
        }
```

5.3 Threads

Java was conceived from the start to be a language that would support multi-threaded programming. Threads are a first class object and the synchronised keyword was provided to assist in co-ordinating threads. However as Java was one of the first to support multiple threads and incorporate them into its design, this meant that the designers did not get its prefect on their first attempt. As the system deals with many threads throughout its workings, from progress bars and breaks to listeners, this was an important issue to address.

Java's shortcomings are highlighted in their failed attempts to provide a safe and effective way of stopping a thread once it had started. The original *java.lang.Thread* class included the methods, *start(), stop()*, *stop(Throwable)* and *suspend()*, *destroy()*, and *resume()*, which were conceived to provide the foundations for the basic functionality for starting and stopping a thread. Out of all these methods, only the *start()* method remains and has not been deprecated.

Stopping threads is not the only problems facing previous versions of Java however, exception handling of what a thread might produce once in its run method were not properly dealt with. This lead to poor coding such as that shown below as it is not easy to distinguish when the thread has terminated with an exception.

```
public void run() {
   // note the Thread.run() method is declared
   // NOT to throw any checked exceptions
   try {
      ....
     // some code here which might throw
     // a checked exception such as an IOException
     ..
   } catch (IOException ioex) {
      ioex.printStackTrace();
   }
}
```

5.3.1 Solution to stopping a thread

After the depreciated of the Thread's stop(), suspend() etc methods, the only solution to safely stopping a thread is to have it exit its run() method. According to Sun Microsystems's article, "Why are Thread.stop, Thread.suspend and Thread.resume Deprecated?"[22] This can be achieved by using a volatile stop flag. This is defined as "blinker" in the code below.

```
private volatile Thread blinker;
public void stop() {
    blinker = null;
}
public void run() {
    Thread thisThread = Thread.currentThread();
    while (blinker == thisThread) {
        try {
            thisThread.sleep(interval);
        } catch (InterruptedException e){
        }
        repaint();
    }
}
```

The volatile keyword is used to make sure that prompt communication between threads is adhered to. If a field is declared volatile it means that a thread must reconcile its working copy of the field with the master copy every time the variable is accessed. Operations on the master copies of one or more volatile variables on behalf of a thread are performed by the main memory in exactly the order that the thread requested. [28]

5.4 Displaying all the stretches in categories

In order to achieve this, the system makes use of regular expressions and searches the stretch type for keywords. The type of stretch is stored in the XML file with all the stretches information. The system stores the stretch types in an Arraylist, then searches through the Arraylist trying to match the stretch types into the pre-defined categories of Eyes, Hands, Legs, Back and Neck. A segment of the code that achieves this is detailed below:

```
…

//for loop. Loops through the Arraylist

    for (int i = 0; i < typeList.size(); i++) {

//gets the i-th element in the Arraylist.

      String type = (String) typeList.get(i);

       boolean t;

//checks to see if the stretch type contains the phrase “eye”

       if(  t = type.matches("(?i).*eye.*")){

//displays the Title of the stretch, if it is an eye stretch, in the GUI’s stretch display area

          GUI.StretchesEyesTextArea.append((String) ReadAndPrintXMLFile.titleList.get(i)+ "\n");

…

//If the stretch is not an eye stretch, it checks against the next category
```

```
//This regular expression looks for either a hand or arm in the stretch type

        }else if (t = type.matches("(?i).*hand.*| *arm.*")){

…
```

Chapter 6
Testing

Testing is a major part of any project and with Aries this was no exception. Throughout the entire development process Aries has been tested thoroughly to ensure that it fulfils its requirements robustly and reliably.

6.1 Incremental Testing

Once the basic foundations of the system had been laid out and thoroughly tested, additional features were added to the system. Once a new feature had been added the system was again thoroughly tested by both myself and a third party. These tests were vigorous and adhered to a detailed test plan covering both the new features and the existing features, ensuring that they interact properly and that no regression has been introduced into the system.

This form of testing guaranteed that any bugs in the system were discovered as early in the development of the system as possible, ensuring that there was adequate time to address any issues that may have arisen.

Many bugs were found using this process and were subsequently corrected. However a few bugs remain in the system (See Section 7.7.3).

6.2 User Interface Testing

The graphical user interface on the Aries system was strenuously tested to ensure that all buttons did what they were supposed to and that under normal circumstances the user would not be able to crash the system through interaction. Many of Aries' commands were implemented using Actions, as a result of this; many only need to be tested once as they are instances of the same class. An example of this is the "Rest Break" menu item and the "Stretch Pop up" button on the main GUI which both use the same Action.

Any input that the system takes from the user is also tested to avoid the system crashing as a result of extreme or unexpected inputted user day. Although the Aries system monitors the user with very little user input, the system does use combo boxes to get when the user wants to take their scheduled breaks. Detailed here in Table 21.

Window	Tested Element	Test	Result
GUI – Settings Tab	Morning Hour Combo Box	Selecting an hour for the morning break – morning break should appear at this hour	PASSED
GUI – Settings Tab	Morning Minute Combo	Selecting an minute for the morning break -morning break	PASSED

	Box	should appear at this minute past the hour	
GUI – Settings Tab	Morning AMPM Combo Box	Selecting an AMPM for the morning break - morning break should appear before or after noon.	PASSED
GUI – Settings Tab	Lunch Hour Combo Box	Selecting an hour for the lunch break – lunch break should appear at this hour	PASSED
GUI – Settings Tab	Lunch Minute Combo Box	Selecting an minute for the lunch break - lunch should appear at this minute past the hour	PASSED
GUI – Settings Tab	Lunch AMPM Combo Box	Selecting an AMPM for the lunch break - lunch break should appear before or after noon.	PASSED
GUI – Settings Tab	End of the Day Hour Combo Box	Selecting an hour for the End of the Day break – End of the Day break should appear at this hour	PASSED
GUI – Settings Tab	End of the Day Minute Combo Box	Selecting an minute for the End of the Day break - End of Day break should appear at this minute past the hour	PASSED
GUI – Settings Tab	End of the Day AMPM Combo Box	Selecting an AMPM for the End of the Day break – End of day break should appear before or after noon.	PASSED
GUI – Settings Tab	System Strictness Slider	Change the sliders value – systems time period between the breaks should change accordingly	PASSED
GUI – Settings Tab	Save Settings Button	The settings show have now taken effect	PASSED
GUI	Posture Guide Button	The Posture Guide should appear	PASSED
GUI	Stretches Pop up Button	The Rest Break window should appear	PASSED
GUI– About Menu item	Help Button	The Help window will appear	PASSED
GUI– About Menu item	RSI – FAQ Button	The RSI – FAQ window will appear	PASSED
GUI – About Menu item	Help Button	The Help window will appear	PASSED
GUI – File Menu	Micro Break Button	A micro Break will appear	PASSED
GUI – File Menu	Quit button	The Aries System will exit	PASSED

GUI	Cross at the top of the window	The Aries system will be hidden	PASSED
Aries icon - Notification Area	Right-click on icon	A pop – up menu will appear	PASSED
Aries icon - Notification Area	Open option	The Aries GUI will appear	PASSED
Aries icon - Notification Area	Exit option	The Aries system will exit	PASSED
MicroBreak Screen	Ignore Button	The MicroBreak will close	PASSED
Rest Break Screen	Close Button	The Rest Break window will close	PASSED
Rest Break Screen	Another Stretch Button	Another Rest Break will appear	PASSED
Scheduled Break Screen	Dismiss Button	The Break alert will disappear	PASSED
Posture Guide	Close Button	The Posture guide will close	PASSED
About Screen	Close Button	The About screen will close	PASSED
RSI – FAQ	Close Button	The RSI -FAQ window will close	PASSED
Help Screen	Close Button	The Help Screen will close	PASSED
GUI	Posture Guide – Keyboard hotkey	The Posture Guide will open	PASSED
GUI	MicroBreak – Keyboard hotkey	A Micro Break will appear	PASSED
GUI	Rest Break – Keyboard hotkey	A Rest Break will appear	PASSED
GUI	Quit – Keyboard hotkey	The Aries system will exit	PASSED
GUI	Help Screen – Keyboard hotkey	The Help screen will appear	PASSED
GUI	About Screen – Keyboard hotkey	The About screen will appear	PASSED
GUI	RSI – FAQ – Keyboard hotkey	The RSI – FAQ screen will appear	PASSED

Table 21 - Interface Testing Results

6.3 Stress Testing

Stress testing was carried out on the system to determine its stability. This form of testing involved pushing the Aries system beyond its normal operational capacity, and observing the results. As the Aries system monitors the user's activities and runs

in the background, the stress testing plan was to try and overload the system's CPU in order to see if the system continues to monitor the user or if the system crashes. With modern computers having extensive memory and processing power this was a rather hard test to fulfil. The system withstood all my attempts to crash it and continued to monitor the user. This is credited to the event hook taking up very little process power.

6.4 User Testing

User testing allows me the developer, to directly observe the user using the Aries system. The user test is one of the most effective means of evaluating the systems usability. The test will highlight any usability problems in real terms and usability can be measured throughout the test by calculating the user's performance.

The test is designed to see how the system interacts with the user, and to discover if the system is lenient enough to allow the user to complete their work whilst being firm enough to prevent RSI. The test will give an indication to whether the system can be tolerated and which functions that it provides are appreciated by the user and which are not.

6.4.1 Test Results

Test subject 1: Colleague (Expert)
Test subject 2: Novice

The first set of tasks were to see how easy it is to navigate through the Aries system's GUI.

Task 1: Bring up a Rest Break.
Subject1: Clicked on the "Stretches Pop-up" button located on the GUI's main window. Although this was not labelled "Rest Break", they took an educated guess and tried it.
Time: A few seconds
Outcome: Success

Subject2: Clicked on the file menu and then click on the "Rest Break" button.
Time: 15 seconds
Outcome: Success

Task 2: Your legs are feeling tired and you decided to do a stretch designed for your legs. Find me such a stretch using the system.

Subject1: Clicked on the "Stretches" tab and then clicked on the "Legs" tab
Time: 10 seconds
Outcome: Success

Subject2: Clicked on the "File" menu, then clicked on the "About" menu, acknowledge the existence of a Help button but carried on. Clicked on the "Stretches" tab on the main GUI then after seeing the "Legs" tab, clicked on that.
Time: 30 seconds
Outcome: Success

Task 3: Set the lunch time break to quarter past one in the afternoon.

Subject1: The user clicked on the settings tab and changed the lunchtime combo boxes to 1:15:PM
Time: 10 seconds
Outcome: Success

Subject2: The user clicked on the settings tab and changed the lunchtime combo boxes to 1:15:PM
Time: 10 seconds
Outcome: Success

Task 4: Set up your workstation ergonomically to help reduce your risk of RSI.

Subject 1: Clicked through the GUI's main tabs before seeing the "Posture Guide" button on the main GUI. Clicked the "Posture Guide" button.
Time: 5 seconds
Outcome: Success

Subject 2: Clicked on the Posture Guide button on the main GUI.
Time: A few seconds
Outcome: Success

Task 5: Quit the system.

Subject 1: Clicked on the cross at the top right of the GUI.
Time: A few seconds
Outcome: Failure

Once I pointed out that the system was still running. They noticed it's icon in the systems notification area and right clicked on it and left clicked on the "Exit" option in the popup menu
New Outcome: Success.

Subject2: Clicked on the "File" menu and then clicked on the "Quit" menu item.
Time: A few seconds
Outcome: Success

The new series of tests are designed to see how the user and the system interact with the users working with the system monitoring them in the background.

Task 6: Check your emails and then compose an email of roughly 500 words.

Subject1: Did the task asked.
Time: Several minutes (under 10minutes)
Outcome: Nothing happened (success)

Subject2: Did the task asked.
Time: 15 minutes
Outcome: Rest Break appeared after 10 minutes (success)

Task 7: Open a text editor and try type till a Micro Break appears

Subject1: Did the task asked.
Time: 5 minutes
Outcome: Micro Break appeared

Subject2: Did the task asked.
Time: 10 minutes
Outcome: Micro Break appeared

Additional notes from users on this task: Maybe set the intensity level of 3 to being a bit less, as this is when the Micro Break appears, because the level of activity needed to be quite extensive in order to reach this level.

After these tasks were completed I asked the test subjects to take my system away and test it.

After a day I asked the test subject if they had encountered any problems and whether the system was a hindrance or a benefit. These are the issues and recommendations to be addressed in the future (Table 22):

Issues Raised	Future Solutions
Stretches Area in GUI is editable	This has now been fixed
More than one Posture Guide can be on the screen at once	Future Solution: Close the window before calling a new one
More than one Rest Break can be on the screen at once	Future Solution: Close the window before calling a new one
The Rest Break can be minimized, thus the user can have several effectively ignore it.	Due to the fact that there is a close button anyway this is not much of an issue. Although in future it may be require to enforce these breaks in order to prevent the user just ignoring them and RSI still occurring.
When lots of Rest Breaks are open (their timers running), the progress bars (threads) cease running until some of the older	Steps to resolve: Only allow the user to have one break open at a time. Or increase the memory that java is allowed

breaks finish their threads.	

Table 22 - Issues Raised from User Testing

User Testing Conclusion: Both test subjects found the system to be very easy to use and that the system was of definite benefit rather than a hindrance.

6.5 Break/Thread Testing

As threads are a main feature of the Aries system, testing of these threads that appear on the break screens and dictate when these breaks will appear, is a necessity. The threads throughout the system were all tested to ensure that they run and terminated when they were meant to. The use of the three tier system architecture, ensured that this was the case and that threads were dealt with in the correct way.

After extensive testing of the system as a whole, no issues were raised with regards to the threads (except for one minor issue, see Known Bugs section in Appendix C for details). Therefore on a whole the testing of threads can be regarded as a success.

6.6 Cross Platform Testing

The Aries system was designed and implemented using Java which is a platform independent language; however the system hook that listens to the user's activities varies on different operating systems. The Aries system hook is currently only functional on Linux systems at present, as this was the more challenging OS to perform such a hook. This choice was enforced by the use of Linux computers in the Computing Department where the Aries system was being developed. Due to time constraints a system hook for Windows was not implemented, but this will be included in Aries version 1.1 to increase the system's portability.

In addition to this, the systems look and feel differs from system to system, as Aries uses the system's look and feel to render its windows. Java offers a standardised look and feel that can be applied to the system but the design decision was chosen to go with the user system's look and feel as this was seen to aid familiarity and as a result made the GUI more user friendly.

Aries was primarily developed on Centos 5 (a Linux environment) using the IDE Netbeans. However with 91% of personal computers being Windows based [8] the Aries system interface was tested on the Windows environment to ensure that the system still functioned as it should (without the event hook). Although with the event hook being the link between the user's activities and the system, this was a very limited test without this functionality.

When the Aries system was run on a Linux environment the results were both successful and conclusive with everything functioning as it is supposed to.

The results from running the Aries system on the Windows environment (including Windows 7, Vista and XP) were successful although the system had limited functionality. All features not relying on the system's event hook worked successfully.

Chapter 7
Evaluation & Future Work

7.1 Realisation of Primary Goals

Aries meets all of the primary goals that were set out at the beginning of the implementation stage. All of the requirements marked as having a priority rating of 1 or 2 in the requirements section have been met, as well as most of the requirements with a priority of 3. The Aries system has currently been implemented up to and including some of the features of Prototype 1.2.

7.2 Realisation of Secondary Goals

Some of the system's secondary goals have been met throughout the implementation of the system, these include: A pre-break and post-break alert noise.

Implementation only reached prototype 1.2 of the Aries system, this was mainly due to time constraints. If future work was to be carried out on the system, prototype 3.0 could realistically be achieved.

Potential difficulties of fulfilling the necessary requirements include the recording of which applications the user has been using. This would require another system hook, similar to the one implemented to listen to the users actions. Further future pitfalls include the details of what the user's activities are, needed to determine what keys are being pressed by the users right hand and left hand. This poses a potential security breach and would be challenging to implement in the Linux environment where security protocols are much tighter than in the Windows environment.

7.3 Evaluation of Completed Work

Aries can be distributed at present, but only to a limited potential client base. These reasons are listed in the following sections.

7.3.1 Improvements to core functionality

A System hook would need to designed and implemented first and foremost if the system is to distribution widely, due simply to the fact that ninety one per cent of computer users worldwide use the windows environment[8].

The introduction of a way to save the user's settings would also greatly improve the system's core functionality. This would ensure that the user would not have to set the time they want to take their breaks throughout the day every time they launch the Aries system. These can be achieved in two ways:

1) Store the user's details in a database, outlined in the section 4 of this report.
2) Store the user's details in an XML file, this would offer a more portable option to that of the database.

The introduction of a means of storing user settings could also be expanded to allow Administrator settings for the system, which would allow a company boss to set the strictness level for his whole work force. Passwords and usernames could be stored in a similar way to user settings.

Whichever option is chosen, the Aries system will need to be updated to accommodate the new functionality.

7.3.2 GUI improvements

Aries' graphical user interface has been extensively tested and is ready for distribution. Several improvements could be made to the GUI however to improve it further, shown here in Table 23.

7.3.2.1 Known GUI Issues

Issue	Description	Priority
RSI – FAQ screen	At present the RSI – FAQ screen information on RSI text area starts with the scroll bar at the bottom rather than the top. Thus meaning that the user has to scroll back up to the top. This could be improved by finding a means of making the scroll bar start at the top and thus at the beginning of the information about RSI text.	Low
Rest Break Screen	If the name or type of the stretch being displayed is longer than the length of the text field then some of the text is cut off. This is a result of the inability to line wrap on a text field within Java. This could be resolved by	Low

	changing the text field to a text area.	

Table 23- Known GUI Issues

7.3.2.2 Usability improvements

A few improvements to the Aries system were observed during the User testing, detailed in section 6.4. The follow is a list of these proposed improvements.

- Put a Help screen button on the main GUI as users tended to give up on tasks rather than check this help guide. The presence of a Help button on the main GUI may prompt the user to check the help guide.
- Support the ability to increase the size of the text on the GUI to accommodate for users with sight problems. Everything else on the GUI including buttons and images should stay in proportion.

7.3.3 Known Bugs

Several bugs where revealed by the extensive testing that the Aries system was exposed to. These bugs are listed below in Table 24.

Bug	Description	Priority
Rest Break Screen	When lots of Rest Breaks are open (their timers running), the progress bars (threads) cease running until some of the older breaks finish their threads. Steps to resolve: Only allow the user to have one break open at a time. Or increase the memory that java is allowed.	Medium/Low
Posture Guide	More than one Posture Guide can be on the screen at once Future Solution: Close the window before calling a new one	Low

Table 24 - Known Bugs

7.4 Future Work

In order to fulfil the initial ambitions of the Aries system future work needs to be carried out on the system. Many of these issues have been highlighted in the sections above, such as making the system fully functional on the Windows OS environment and employing the ability to save the users setting for future sessions.

The system has been designed and implemented to help aid any further development. The use of the three-tier architecture ensures that any changes or additions can be easily implemented without affecting the systems core functionality.

The use of modularity and extensive commenting throughout the source code also ensure that any future work by developers outside the original Aries development team will not be hindered as the code is both easy to follow and understand.

Several additional features may be added to the Aries system that were not in the original system's requirements. A suggestion from the focus group recommended that the Aries system monitor the number of hours that were spent on certain projects. This was not considered as it was a relatively niche addition to functionality. However the idea to make the Aries system have the ability to have plug-ins written for it and have it tailored to a company's needs is one that can be fulfilled. This form of implementation model has been a highly successful one, used by many open source projects, such as Mozilla Firefox [9].

The introduction of additional functionality that would allow the user to view their statistics and activities in the form of graphs and charts would be a good feature. With the introduction of a method of storing the users statistics from previous sessions allowing the user to view and analyse their previous actions. From this data, trends could be discovered and the system tailored and altered to improve the system's effectiveness to combat RSI.

The functionality to support more languages would make the Aries system accessible to a wider spectrum of potential clients and increase the system's usability.

Chapter 8
Conclusions

Overall the Aries project can be labelled a success, with most of functional requirements fulfilled and all of the non-functional and core requirements being fulfilled.

Aries is still however, a work in progress and needs some additional work before it can be widely distributed as outlined in the section 7.4.

All new features added to the Aries system must be exposed to the extensive testing that the current functionality has been put through.

The next step would be to release Aries as a beta product to get further and extensive testing of the system in the public domain. Feedback could then be used to help improve the system and increase both its robustness and reliability.

The Aries project has been a worthwhile undertaking in my opinion and has increased both my knowledge and awareness of the potential threats of RSI. If developed further, I feel that this product could both help and educate computer users worldwide.

The future for Aries' development would be the transition to either an Open Source project, which would be freely available to use, aimed at personal users or as a marketable product aimed at companies.

References

[1] RSIAwareness. 2007. *What are musculo-skeletal disorders?* (Article). [Online]
Available at: http://www.rsi.org.uk/ [Accessed 4/3/2010].
[2] Daily Mail Reporter. 2008. Two thirds of office staff suffer from repetitive strain injury. (Article). [Online]
Available at: http://www.dailymail.co.uk/health/article-1024097/Two-thirds-office-staff-suffer-Repetitive-Strain-Injury.html [Accessed 6/3/2010]
[3] Unknown. Britain at work (Website) [Online]
Available at: http://www.tuc.org.uk/ [Accessed 6/3/2010]
[4] Buckle and Devereux. 1999. European Institute of Health and Medical Sciences, University of Surrey
[5] Buckle. 1999. European Institute of Health and Medical Sciences, University of Surrey
[6] Schneider. 1998. Unknown. [Online]
Available at: http://www.rsi.org.uk/
[7] Mark Cully et al, 1999
[8] Website. 2010. Global Web Stats. [Online]
Available at: http://www.w3counter.com/globalstats.php
[9] Website. 2010. Mozilla Add-ons. [Online]
Available at: https://addons.mozilla.org/en-US/firefox/
[10] Website. 2008. RSI Solutions. [Online]
Available at: http://www.rsi-solutions.co.uk/
[11] Office of Communications and Public Liaison. 2002. "Carpal Tunnel Syndrome Fact Sheet," NINDS. [Online]
Available at:
http://www.ninds.nih.gov/disorders/carpal_tunnel/detail_carpal_tunnel.htm
[12] Website. 2009. Tendonitis and Tenosynovitis. [Online]
Available at: http://www.patient.co.uk/showdoc/23069141/
[13] “NetDoctor”. 2007. What is De Quervain's syndrome? [Online]
Available at: http://www.netdoctor.co.uk/ate/musclesjoints/203187.html

[14] BSSH. 2008. De Quervain's syndrome. [Online]
Available at:
http://www.bssh.ac.uk/patients/commonhandconditions/dequervainssyndrome
[15] “Editor”. 2008. Thoracic Outlet Syndrome (TOS). [Online]
Available at:
http://www.medicinenet.com/thoracic_outlet_syndrome/article.htm
[16] BSSH. 2008. Trigger finger/thumb. [Online]
Available at:
http://www.bssh.ac.uk/patients/commonhandconditions/triggerdigits
[17] Unknown. 2008. Intersection syndrome. [Online]
Available at: http://en.wikipedia.org/wiki/Intersection_syndrome
[18] Website.2009. Reflex Sympathetic Dystrophy Syndrome (RSDS). [Online]
Available at:
http://www.medicinenet.com/reflex_sympathetic_dystrophy_syndrome/article.htm
[19] Kadmi. 2010. Break Time. [Online]

Available at: http://www.kadmi.com/products.html
[20] Cheqsoft. 2000. Break Reminder. [Online]
Available at: http://www.cheqsoft.com/break.html
[21] Wellnomics. 2010. WorkPace. [Online]
Available at: http://www.workpace.com/
[22] Albion. 2010. StopNow! [Online]
Available at: http://www.albionresearch.com/astopnow/
[23] Atawell. 2010. Workplace Angel. [Online]
Available at: http://wpangel.com/
[24] Sphere 10 Software. 2010. RSI Warrior. [Online]
Available at: http://rsiwarrior.com/
[25] Martin. 2009. Eye Defender. [Online]
Available at: http://www.ghacks.net/2008/08/15/eye-defender-reminds-you-to-take-breaks/
[26] 'ThreadBuilder'. 2009. Stress Buster.
Available at:
http://www.freedownloadscenter.com/Business/Time_and_Clock_Tools/Stress_Buster.html
[27] Sun Microsystems. 1999. Why Are Thread.stop, Thread.suspend, Thread.resume and Runtime.runFinalizersOnExit Deprecated? [Online]
Available at:
http://java.sun.com/j2se/1.4.2/docs/guide/misc/threadPrimitiveDeprecation.html
[28] Suzette Pelouch. 2000. Volatile Fields. [Online]
Available at:
http://java.sun.com/docs/books/jls/second_edition/html/classes.doc.html#36930

Appendix A
Aries User Manual

Contents

1 Opening Aries 2
1.1 SplashScreen 2
2 The Aries System 2
2.1 Posture Guide 2
2.2 Aries Overview 4
2.3 Setting the times for the scheduled breaks 4
2.4 Set the Strictness of the Aries system 6
2.5 Rest Breaks 7
2.6 Micro Breaks 9
2.7 Actions 11
2.8 Usage 11
2.9 Stretches 12
2.10 Windows 13
2.11 About Screen 13
2.12 RSI – Frequently Asked Questions 15
2.13 Help 17
2.14 Aries running in the background 18
2.15 Quitting Aries 19

1 Opening Aries

Before you try to launch Aries, please make sure you have Java installed as outlined in Appendix C

To launch the Aries System, Double-click on the Aries .jar file (shown in Figure 26) which can be located in the installation package.

Alternative method:

1) Right-click on the Aries.jar file (Figure 26)
2) Left click on the "Open" option from the resulting pop-up menu.

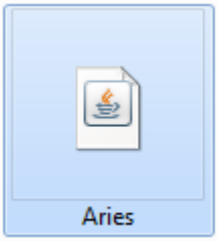


Figure 26 - The Aries .jar file

If Aries fails to load:
Open the Terminal (UNIX, Mac) or the Command Prompt (Windows).
Change the directory using the "cd" command to the directory containing the Aries.jar.
Then type: "java –jar Aries.jar" and press "Enter".

1.1 SplashScreen

You will now be presented with the system's SplashScreen (illustrated below in Figure 27). This screen is displayed whilst the system is loading.

Figure 27 - Aries SplashScreen

2 The Aries System

2.1 Posture Guide

Once the Aries system has loaded, it will immediately begin to monitor your key presses and mouse clicks in order to help you from developing RSI. (NB. No sensitive data will be stored by this system). You will then be presented with the system's Posture Guide (shown in

Figure 28) that shows how your workstation should be best set up to help reduce and prevent the effects of RSI.

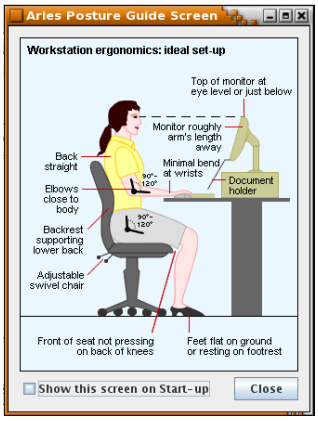

Aries Posture Guide Screen
Workstation ergonomics: ideal set-up
Top of monitor at eye level or just below
Monitor roughly arm's length away
Minimal bend at wrists
Document holder
Back straight
Elbows close to body
Backrest supporting lower back
Adjustable swivel chair
90°-120°
Front of seat not pressing on back of knees
Feet flat on ground or resting on footrest
Show this screen on Start-up
Close

Figure 28 - Aries Posture Guide Screen

To close this guide, left-click either the "Close" button (highlighted below in Figure 29)

Alternative: Left-click on the cross at the top right of the window.

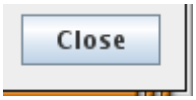


Figure 29 -Close Button

You are able open the Posture Guide at any time whilst using Aries.

One way to do this is to select it from the File menu.

1) Left-click on the "File" menu, located at the top left of the window and shown in Figure 40.

A drop down menu will appear as shown in Figure 41.

2) Click on the Posture Guide Button shown in Figure 30 below.

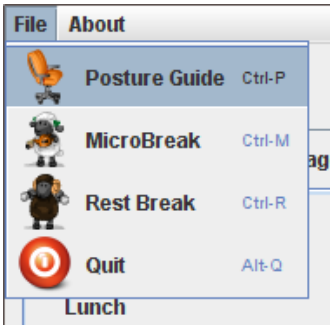


Figure 30 - Highlighted Posture Guide Menu Item

You can also access the Posture Guide screen using a keyboard short key.

1) When Aries is open, press Control-P, to display the Posture Guide

The Posture Guide will now appear.

2.2 Aries Overview

You will then be presented with the system's main graphical user interface. An overview of the system features are be shown in Figure 31.

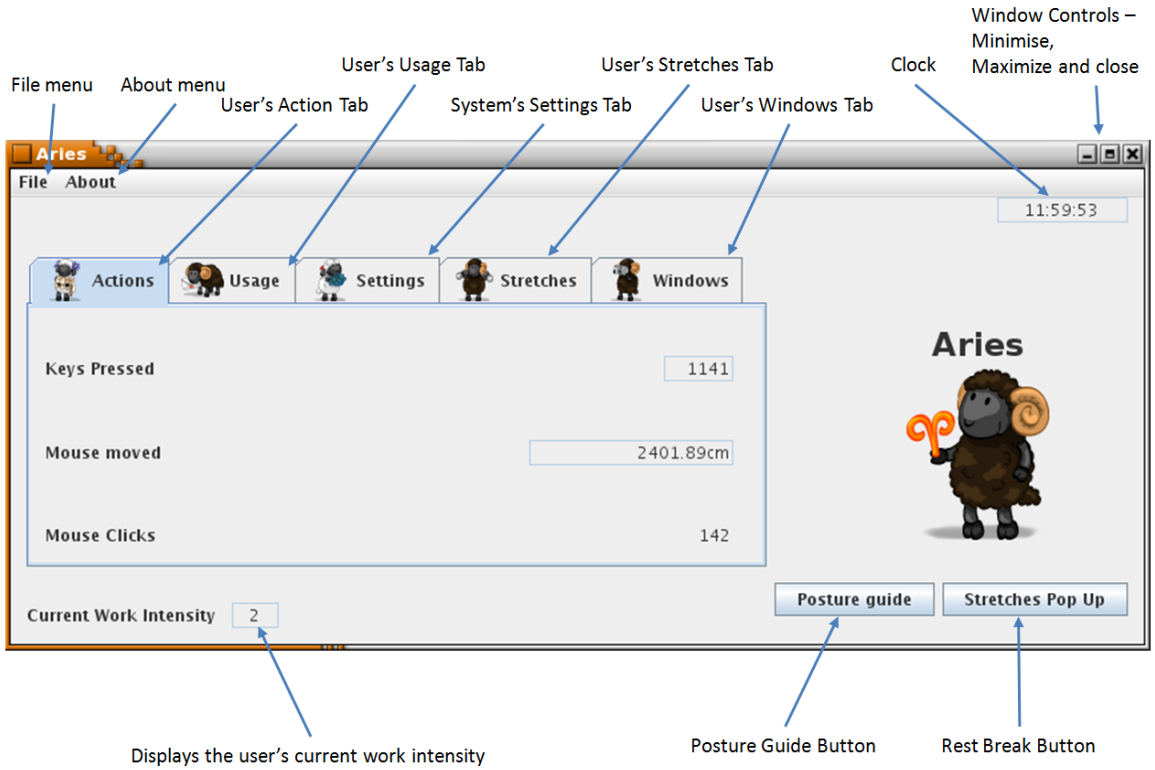


Figure 31 - Aries Overview

2.3 Setting the times for the scheduled breaks

You can set the times that they wish to take your breaks throughout the day. The system allows for you to set reminders for three breaks. These are a morning break, lunch break and when it is the end of the day.

1) Locate and left-click on the “Settings” Tab (it is the third one) and then you will be presented with the settings screen (illustrated in Figure 32).

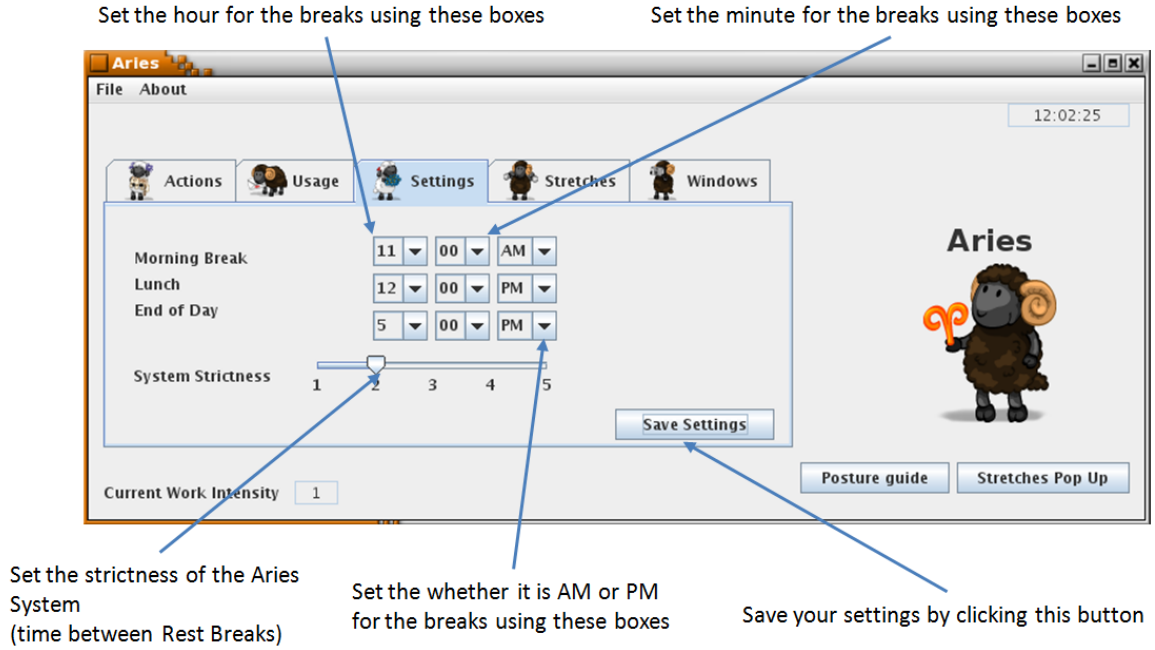


Figure 32 - Aries Settings Screen

2) Left-click on the pull down box that you want to change. A drop down list will appear (shown in Figure 33 below)

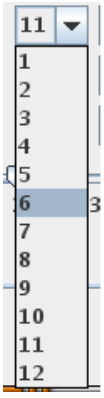


Figure 33 - Scheduled Break Hour Drop Down

3) Left-click on your desired value then make sure that the AM/PM box (shown below in Figure 34) is in the correct state.

Figure 34 - AM/PM Box

Left-click on your desired value and then click on the “Save Settings” button (shown below in Figure 35)

Figure 35 - Save Settings Button

When the Aries’ clock reaches one of the times that have been set by the user, the corresponding break screen will appear. These break screen are shown below in Figure 36.

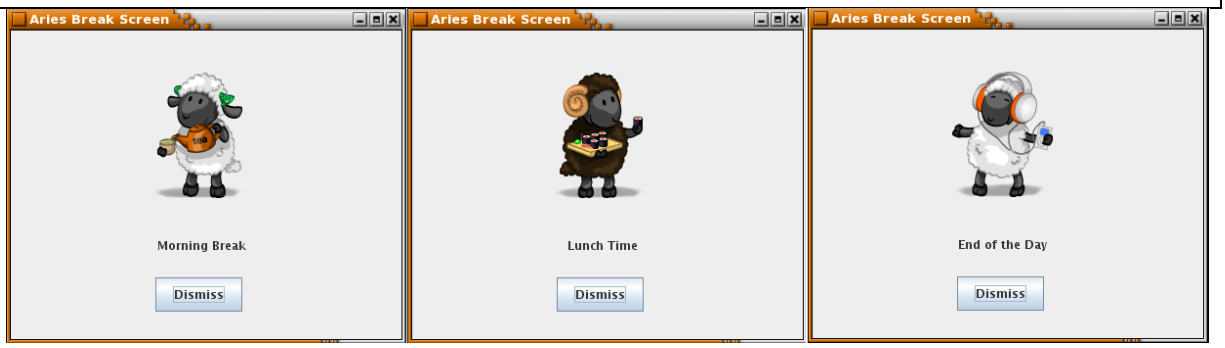


Figure 36 - Aries Scheduled Break Screens

You can then dismiss the break alert.

Left-click on the “Dismiss” button located in the lower middle of the window and highlighted in Figure 37.

Alternative: Left-click the cross located at the top right of the screen.

Figure 37 - Scheduled Break Dismiss Button

2.4 Set the Strictness of the Aries system

1) Locate and click on the “Settings” Tab (it is the third one) and then you will be presented with the settings screen (illustrated in Figure 32).

2) Adjust the strictness slider (shown in Figure 38)

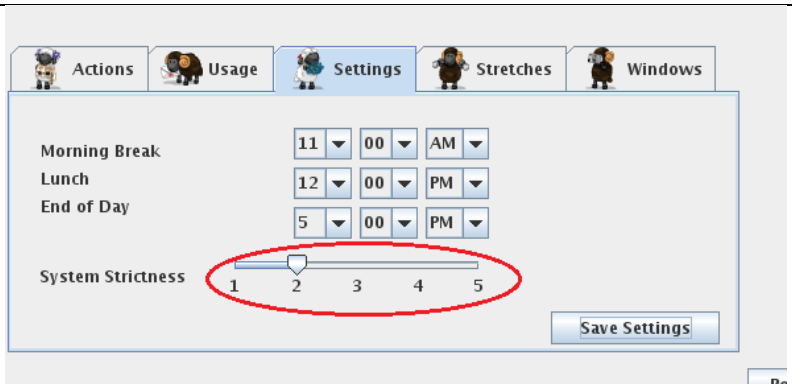


Figure 38 - Strictness Slider

The system strictness slider alters the time between the Rest Breaks. The settings are as follows:
Setting 1: 7 minutes
Setting 2: 12minutes
Setting 3: 17minutes
Setting 4: 20minutes
Setting 5: 25minutes

3) Once you have the slider the desired value, click on the "Save Settings" button (shown in Figure 35)

These new settings will take place once the current break time has passed (this can be found under the "Usage" tab and is shown in Figure 52, Aries default is strictness setting is 1, so will encounter Rest Breaks every 6.66 minutes until you have saved your new settings.

2.5 Rest Breaks

Aries Rest Breaks consist of a RSI preventing stretch along with a description of what areas the stretch benefits and how to perform the stretch (shown in Figure 43). These stretches appear at predetermined intervals, highlighted above in section 2.4. You can however view such a break at anytime.

1) Left-click on the "Stretches Pop Up" button (shown below in Figure 39) on Aries Interface.

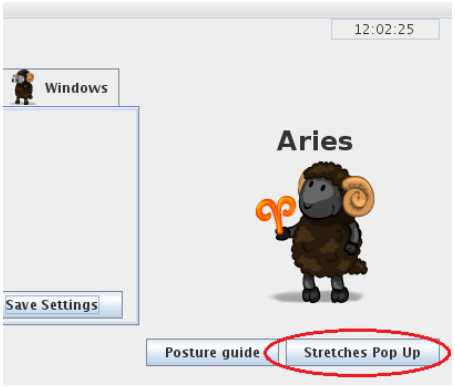


Figure 39 - Stretches Pop Up Button

Another way to take a Rest break is to select it from the File menu.

Alternative:

1) Left-click on the “File” menu, located at the top left of the window and shown in Figure 40 below.

Figure 40 - File Menu Button

A drop down menu will appear as shown in Figure 41.

Figure 41 – File Menu

2) Left-click on the “Rest Break” Button shown in Figure 42.

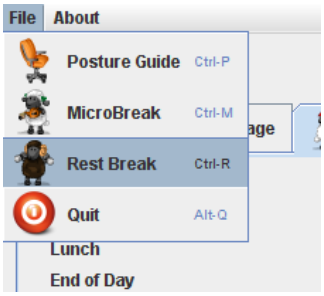


Figure 42 – Highlighted Rest Break Menu Item

You can also access the Rest Break screen using a keyboard short key.

1) When Aries is open, press Control-R, to display a Rest Break.

The rest break will then appear.

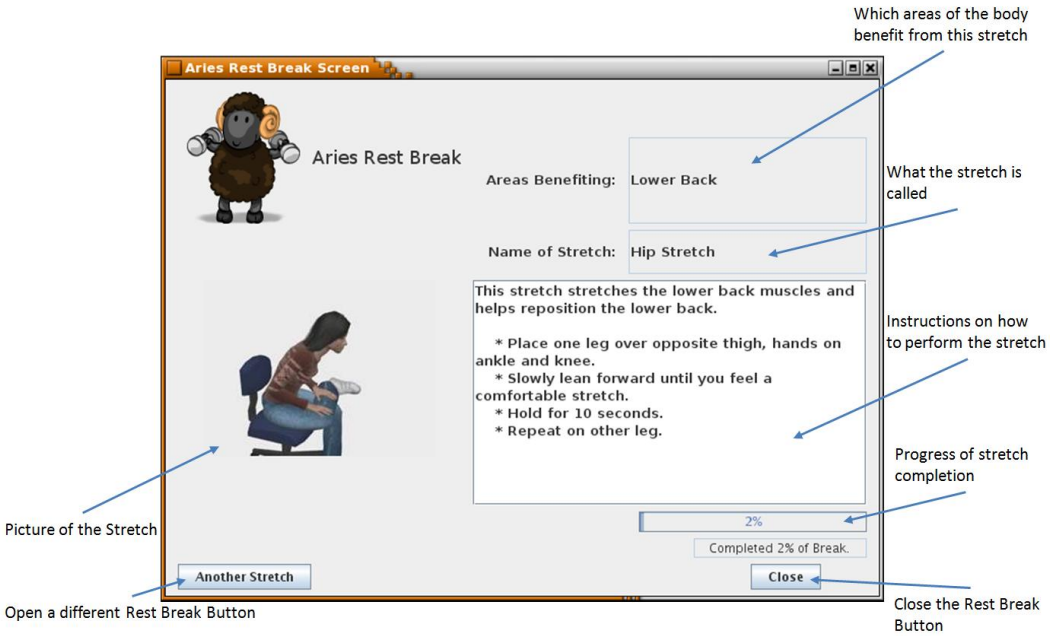

Which areas of the body benefit from this stretch
Aries Rest Break Screen
Aries Rest Break
Areas Benefiting:
Lower Back
What the stretch is called
Name of Stretch:
Hip Stretch
This stretch stretches the lower back muscles and helps reposition the lower back.
* Place one leg over opposite thigh, hands on ankle and knee.
* Slowly lean forward until you feel a comfortable stretch.
* Hold for 10 seconds.
* Repeat on other leg.
Instructions on how to perform the stretch
Progress of stretch completion
2%
Completed 2% of Break.
Picture of the Stretch
Another Stretch
Close
Open a different Rest Break Button
Close the Rest Break Button

Figure 43 - Aries Rest Break Screen

You can then choose to view another stretch if they do not visit to do the one shown.

1) Left-click on the “Another Stretch” Button located at the bottom right of the window and highlighted in Figure 44.

Figure 44 - Highlighted Another Stretch Button

This will display a new Rest Break.
You can also close the Rest Break before it is scheduled to finish.

1) Click on the “Close” button located at the bottom right of the window and highlighted in figure 2.5.7.

Alternative:

1) Left-click the cross located at the top right of the screen.

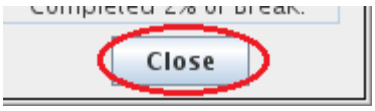


Figure 45 - Highlighted Close Button

2.6 Micro Breaks

Aries’ Micro Breaks encourage you to stop what they are doing for a short period of time. These micro breaks will appear depending on your work intensity. When your work intensity reaches 3 or above, as illustrated in Figure 46, a Micro Break (Figure 47) will appear.

Figure 46 - Work Intensity Field

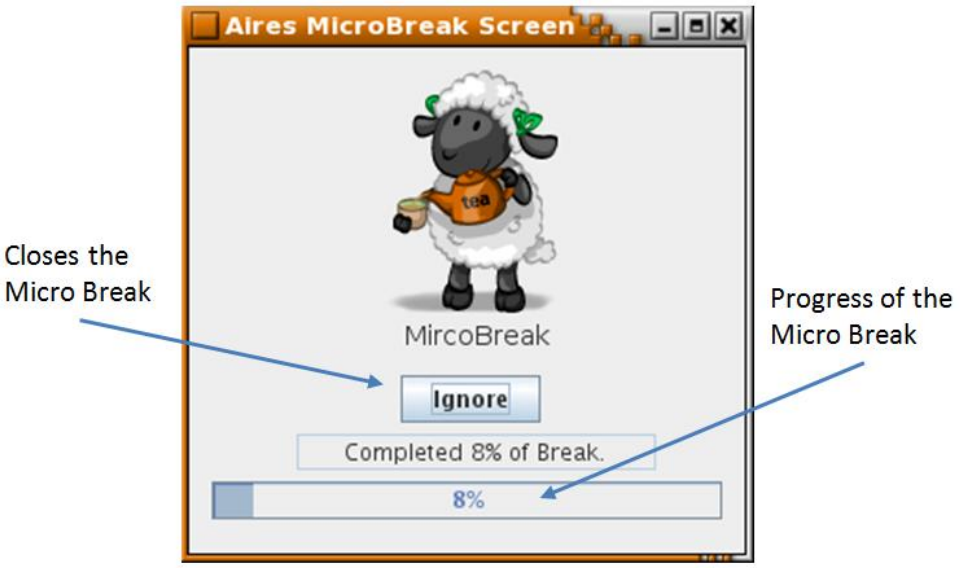


Figure 47 - Micro Break Screen

Another way to take a Micro Break is to select it from the File menu.

1) Left-click on the "File" menu, located at the top left of the window and shown in Figure 40.

A drop down menu will appear as shown in Figure 41.

2) Left-click on the Micro Break Button shown in Figure 48.

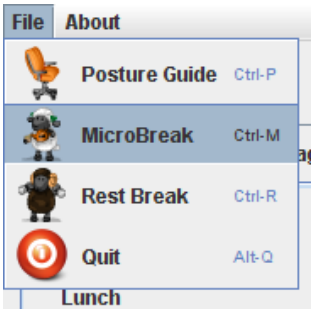


Figure 48 - Highlighted MicroBreak Menu Item

You can also access the Micro Break screen using a keyboard short key.

1) When Aries is open, press Control-M, to display a Rest Break.

The Micro Break will then appear.
You can close the Micro Break before it is scheduled to finish.

1) Left-click on the "Ignore" button located in the middle of the window and highlighted in Figure 49.

Alternative:

1) Left-click the cross located at the top right of the screen.

Figure 49 - Highlighted Ignore Button

2.7 Actions

The Actions tab is the first of the tabs on Aries' Graphical User Interface shown here is figure 2.7.1.

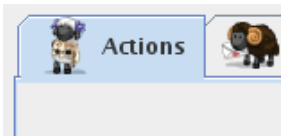


Figure 50 - Highlighted Ignore Button

This tab shows various statistics about your activities since running Aries. These activities include number of keys pressed, number of times the mouse has been clicked and the distance in centimetres that the mouse has travelled. These features are illustrated in Figure 51 below.

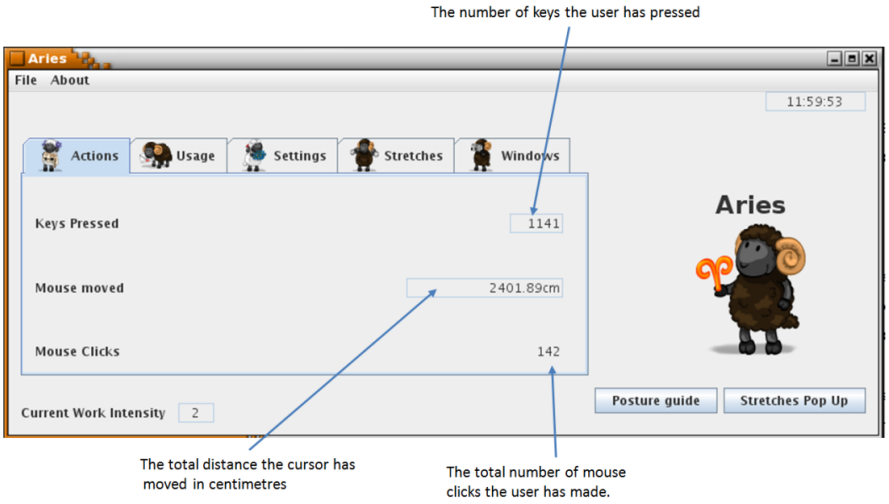


Figure 51 - Aries Actions Tab

2.8 Usage

The Usage tab is the second of the tabs on Aries' Graphical User Interface shown here is Figure 52.

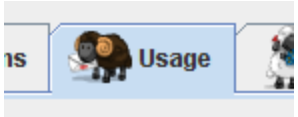


Figure 52 - Usage Tab

The Usage tab shows the time remaining till the next Rest Break in the form of a progress bar. The tab also shows the total time you have been using the Aries system for. These features are highlighted in Figure 53 below.

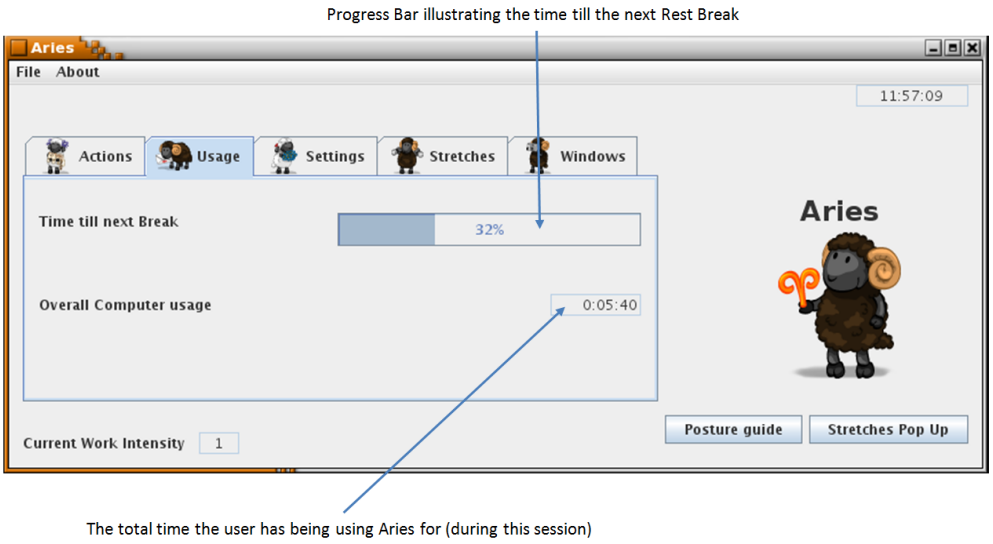


Figure 53 – Aries Usage Tab

2.9 Stretches

The Stretches tab is the fourth of the tabs on Aries' Graphical User Interface shown here is Figure 54.

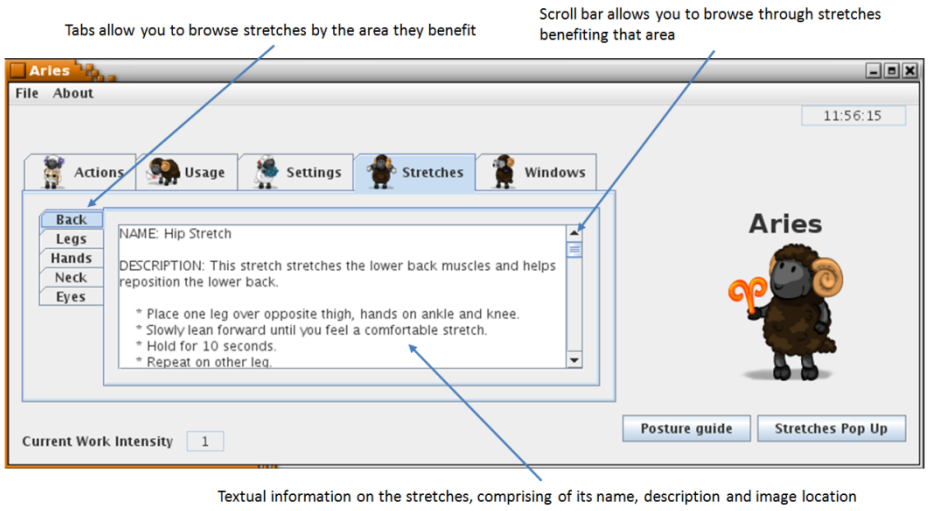

Tabs allow you to browse stretches by the area they benefit
Scroll bar allows you to browse through stretches benefiting that area
Aries
File About
11:56:15
Actions
Usage
Settings
Stretches
Windows
Back
Legs
Hands
Neck
Eyes
NAME: Hip Stretch
DESCRIPTION: This stretch stretches the lower back muscles and helps reposition the lower back.
* Place one leg over opposite thigh, hands on ankle and knee.
* Slowly lean forward until you feel a comfortable stretch.
* Hold for 10 seconds.
* Repeat on other leg.
Aries
Current Work Intensity
1
Posture guide
Stretches Pop Up
Textual information on the stretches, comprising of its name, description and image location

Figure 54 - Aries Stretches Tab

You can browse the stretches by which area of the body that you aim to help protect from RSI.

1) Left-click on the desired tab on the left hand side (highlighted below in Figure 55).

The text will now change in the text area to correspond to this action.
If there are several stretches that aim to protect the same area of the body from RSI you may need to scroll down to find a stretch that they wish to perform.

2) Click on either the up arrow or the down arrow on the scroll bar (which is highlighted below in Figure 55).

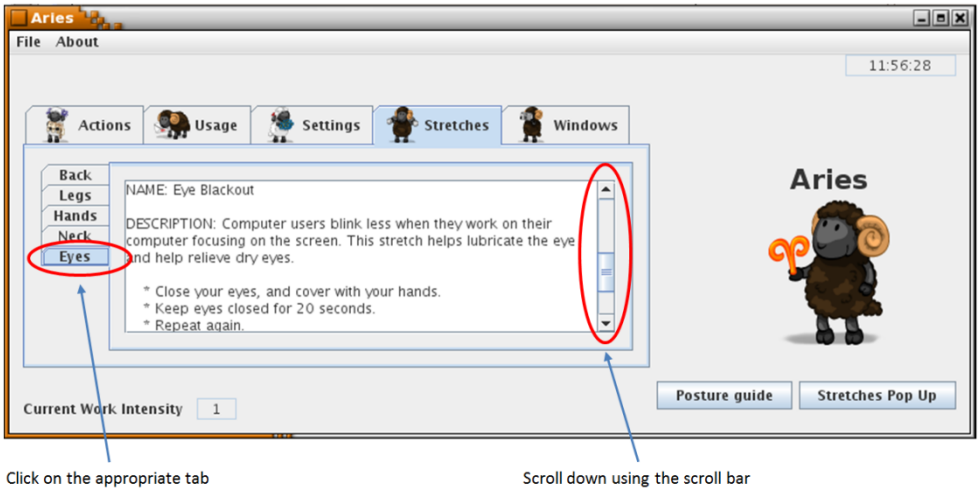


Figure 55 - Highlighted Stretches Tab

2.10 Windows

The windows tab (shown here in Figure 56) was not able to be implemented in time for the Aries version 1.0 so will appear in version 1.1. The tab however remained present so that the Aries layout would not change and you would not need to re familiarise themselves with the system.

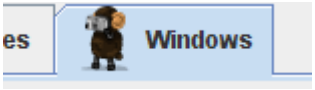


Figure 56 - Aries Windows Tab

2.11 About Screen

The About Screen outlines the motivations behind the Aries System. The About Screen can be accessed by clicking on the About Menu.

1) Left-click on the About Menu (highlighted in Figure 57 below)

Figure 57 - About Menu

The following drop down menu will appear, shown here in Figure 58.

Figure 58 - About Menu drop down

2) Left-click on the About Menu item shown below in Figure 59

Figure 59 - Highlighted About Menu Item

You can also access the About screen using a keyboard short key.

1) When Aries is open, press Control-A, to display the About screen.

The About Screen will then appear. This screen is shown here in Figure 60.

Aries About Screen
RSI Prevent
RSI Prevent is Software designed and developed to help reduce RSI in the workplace.
This system was designed and implemented by Peter Kindness In the Java langauge, as part of his 4th Year Computing Honours Project at Aberdeen University.
Close

Figure 60 - Aries About Screen

To close the About screen.

1) Left-click on the "Close" button located at the bottom right of the window and highlighted in Figure 61.

Alternative:

1) Left-click the cross located at the top right of the screen.

Figure 61 - Highlighted Close Button

2.12 RSI – Frequently Asked Questions

Aries's frequently asked questions screen about RSI offers you answers to many of the questions you may be wondering about RSI. To access this screen:

1) Left-click on the About Menu (highlighted in Figure 57)

A drop down menu will appear, shown in Figure 58.

2) Left-click on the RSI – FAQ Menu item highlighted in Figure 62.

Figure 62- Highlighted RSI - FAQ Menu item

You can also access the RSI - FAQ screen using a keyboard short key.

1) When Aries is open, press Control-Q, to display the RSI - FAQ screen.

The RSI - FAQ Screen will then appear. This screen is shown here in Figure 63.

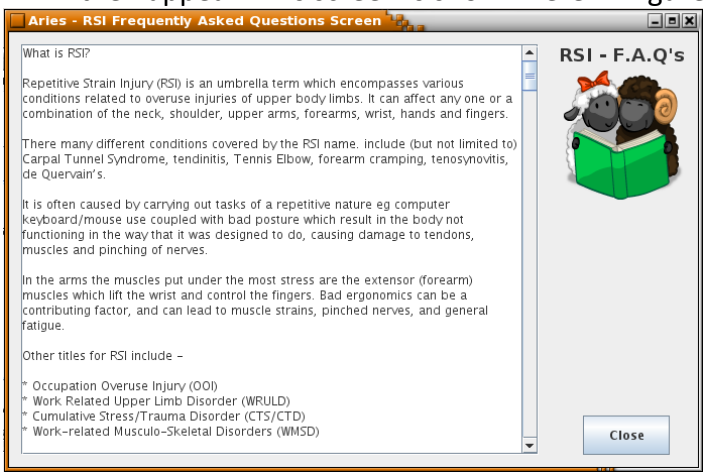

Aries - RSI Frequently Asked Questions Screen
What is RSI?
Repetitive Strain Injury (RSI) is an umbrella term which encompasses various conditions related to overuse injuries of upper body limbs. It can affect any one or a combination of the neck, shoulder, upper arms, forearms, wrist, hands and fingers.
There many different conditions covered by the RSI name. include (but not limited to) Carpal Tunnel Syndrome, tendinitis, Tennis Elbow, forearm cramping, tenosynovitis, de Quervain's.
It is often caused by carrying out tasks of a repetitive nature eg computer keyboard/mouse use coupled with bad posture which result in the body not functioning in the way that it was designed to do, causing damage to tendons, muscles and pinching of nerves.
In the arms the muscles put under the most stress are the extensor (forearm) muscles which lift the wrist and control the fingers. Bad ergonomics can be a contributing factor, and can lead to muscle strains, pinched nerves, and general fatigue.
Other titles for RSI include –
* Occupation Overuse Injury (OOI)
* Work Related Upper Limb Disorder (WRULD)
* Cumulative Stress/Trauma Disorder (CTS/CTD)
* Work-related Musculo-Skeletal Disorders (WMSD)
RSI - F.A.Q's
Close

Figure 63 - Aries RSI - FAQ Screen

The RSI – FAQ screen consists of a large text area that contains information and answers to frequently asked questions about RSI. You may want to scroll through these questions to find the answer they desire. To do this:

1) Left-click on either the up arrow or the down arrow on the scroll bar (which is highlighted below in Figure 64).

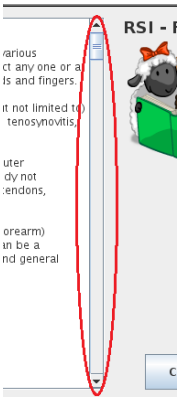


Figure 64 - RSI - FAQ Screen Scroll bar

To close the RSI - FAQ screen.

1) Left-click on the "Close" button located at the bottom right of the window and highlighted in Figure 65.

Alternative:

1) Left-click the cross located at the top right of the screen.

Figure 65 - RSI - FAQ Close Button

2.13 Help

Aries's help screen is a guide to help you with the functionality that they system provides.

To open the help guide shown here in Figure 66.

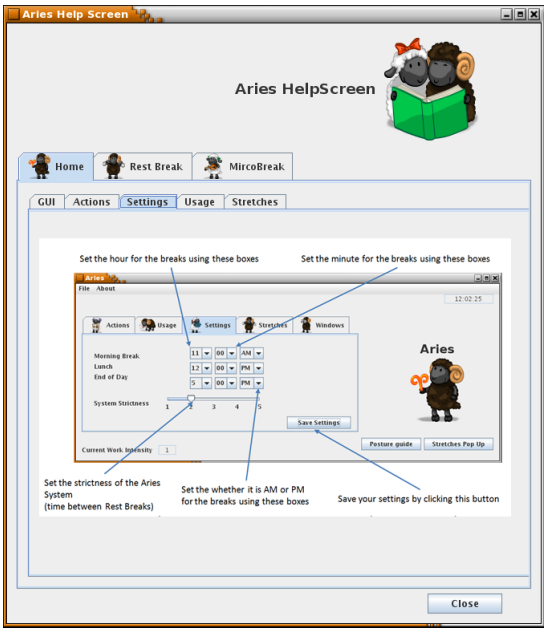


Figure 66 - Aries Help Screen

1) Left-click on the About menu (shown in Figure 57)

2) Left-click on the Help button (shown below in Figure 67)

Figure 67 - Highlighted Help Menu Item

You can also access the Help screen using a keyboard short key.

1) When Aries is open, press Control-H, to display the Help screen.

The help screen will now appear.

To navigate through the help screen click on the tabs located at the top of the screen and highlighted in Figure 68.

Figure 68 - Highlight Help Tabs

To close the Help screen:

1) Left-click on the "Close" button located at the bottom right of the window and highlighted in Figure 69.

Alternative:

2) Left-click the cross located at the top right of the screen.

The help screen will now close.

Figure 69 – Help Screen Close Button

2.14 Aries running in the background

If you close the Aries interface by clicking on the cross at the top right of the screen the system will continue to monitor you in the background.

When the Aries is launched, it will appear in the systems notification area (shown here in Figure 70).

Figure 70 - Aries Icon in the Notification Area (Linux and Windows)

To reopen the Aries application:

1) Right-click on the Aries icon show in Figure 70.

The pop up shown here in Figure 71 will appear.

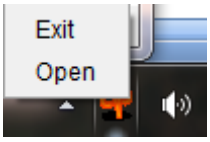


Figure 71 - Notification Icon Pop up

2) Left-click on the “Open” option in the popup menu

The main Aires interface should now appear.

2.15 Quitting Aries

To quit the Aries Application

1) Left-click on the File Menu shown previously in Figure 40.

2) Left-click on the Quit Menu item shown below in Figure 72.

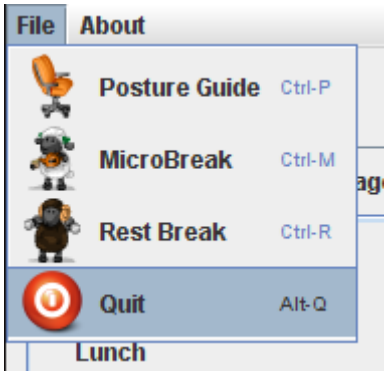


Figure 72 - Quit Menu Item

You can also quit the application at anytime using a keyboard short key.

1) When Aries is open, press Alt-Q, to exit the Aries application.

If Aries interface is closed you can still terminate the application.

1) Right-click on the Aries icon show in Figure 70.

The pop up shown in Figure 71 will appear.

2) Click on the “Exit” option in the popup menu

The Aries system will now close.

Appendix B
Aries Maintenance Manual

Contents

1. Requirements and Dependencies....2
2. Building....2
3. Installation guide....2
4. Source Code Structure....3
4.1 Aries.GUI....3
4.2 Aries.Logic....3
4.2.1 Aries.Logic.EventHook....4
4.3 Stretches.Text....4
5. Known Bugs....4
6. Instructions for possible forms of Maintenance....5
6.1 Add a new stretch....5
6.2. Add another Scheduled Break Screen....5

1. Requirements and Dependencies

The Aries application requires:

Description	Required	Notes
Operating System	Any supporting Java SE 6 Runtime	Please note that the Aries system version 1.0 will only be able to monitor the user on the Linux environment. Version 1.2 will extend this functionality to other OS's like Windows.
Hard Disk Space	4MB	
Memory	256MB	
Java Version	6 SE	Or above. See Appendix C to install
Packages	Libhook .commons	The Libhook library is bundled in the 'EventHook' folder within the 'Logic' folder. The .commons library is stored in the 'dist' folder.
Additional Requirements	XML file containing all the system's stretches	Bundled with the Aries application

Table 25 - Aries Requirements and Dependencies

2. Building

All the source code is provided in the src subfolder within the Aries folder in the archive. Aries requires the libhook library, which is in the EventHook subfolder within the Logic folder.

The system also requires the Stretches XML file that contains all the information about the stretches used by Aries. This file is bundled within the Aries package. All the graphics needed by the system are also stored within the application within the Graphics folder.

The system also requires the .commons library. This library is stored in the 'dist' subfolder in the Aries directory.

To build using Linux:

1) Open the 'dist' folder in the Aries directory.
2) Right-click on the Aries.jar file and left-click 'Open in archive".
3) Left-click and drag the 'Org' folder from the 'dist' folder into the archive window.
4) Double click on the Aries.jar file.
5) The system will now open.

You can also import the 'src' folder into an IDE such as Netbeans or Eclipse.

3. Installation guide

1. Extract the Aries.jar file from the Aries compressed archive
    a. On Linux, right click and click "Extract contents" – using gunzip or ark

b. On Windows, right click and click “Extract all…” – using 7-Zip or similar

2. Double click on the Aries.jar file to run the application

4. Source Code Structure

4.1 Aries.GUI

Class Name	Description
About Screen.java	The window displayed to the user when they access the About screen
Break.java	The window displayed to the user when it is time for a scheduled break
BreakWrapper.java	Extends the Break class and passes the correct image and alert type
FAQ.java	The window displayed to the user when they access the RSI – FAQ screen
GUI.java	Aries main graphical user interface
GuiWrapper.java	Extends the GUI and deals with all inputs, such as eventlisteners, and initiates a response by making calls on other classes. Orchestrates the GUI and other classes to perform actions based on input.
Help.java	The window displayed to the user when they access the Help screen
MicroBreakScreen.java	The window displayed to the user during a micro break
MicroBreakWrapper.java	Extends the MicroBreakScreen and handles the threads and events
Posture.java	The window displayed to the user when they access the Posture Guide
SplashScreen.java	The window displayed to the user when they first launch Aries
StretchesScreen.java	The window displayed to the user when they encounter a Rest Break
StretchesScreenWrapper.java	Extends the StretchesScreen and deals with displaying the stretch image and instructions along with threads and eventlisteners for the screen.

4.2 Aries.Logic

DisplayTime.java	Deals with and handles the system’s time
Initialiser.java	This is the first class that is called. It is the systems "main" class. Calls the SplashScreen then initialises the GUI
MicroBreak.java	Generates the threads for the MircoBreaks
MinimizeToTray.java	Offers the ability to minimize Aries to the systems Notification Area
NextBreakProgressBar.java	Generates the thread for when the next Rest Break should

	be
SetWorkIntesity.java	Calculates what the users work intensity
DisplayScheduledBreaks.java	Checks to see if a scheduled break needs to be shown
GetSettings.java	Retrieves and sets the users settings

4.2.1 Aries.Logic.EventHook

EventHook.java	This class is a wrapper for the C library
EventHook.class	Is the compiled version of the EventHook.java class
Logic_EventHook_EventHook.c	C code that listens to the users actions
Logic_EventHook_EventHook.h	Header for class Logic_EventHook_EventHook (machine generated)
libhook.so	Library that allows the C code to be intergraded into Java

4.3 Stretches.Text

BreakProgressBar	Handles the progress bar on the Rest Break
DisplayAStretch	Handles the displaying of a Stretch on the Rest Break
DisplayAllStretches	Displays all the stretches in the GUI Stretches tab
ReadAndPrintXMLFile	Parses the XML file and reads the Stretches
Stretches	Used to return values for the parser

5. Known Bugs

Bug	Description	Priority
Progress bars on Rest Breaks and MicroBreaks	The progress bars on these screens delay their starting if the user has opened lots of breaks at the same time. The progress bars will only start when ones open prior have ceased running. The user however is likely to open up multiple breaks at the same time, and as this only occurs when more than ten such breaks or open, it is of low priority. Steps to resolve: Only allow the user to have one break open at a time. Or increase the memory that java is allowed.	Low

6. Instructions for possible forms of Maintenance

6.1 Add a new stretch

To add a new stretch to the system the user/developer should follow the following steps:
Difficulty: Low

1) Open the stretches.xml file located inside the Aries folder.
    a. Double click on the stretches.xml file
        i. Alternatively, right-click on the file and left-click on the "open-with" option
        ii. Click on a suitable editing program such as Notepad.
2) Enter the following template to at the bottom of the file in between the </stretch> tag and the </book> tag.

```
<stretch>

<type>Enter what area your stretch benefits here (Leg, Hands, Eyes,
Back, Neck)</type>
<name>Enter the name of your Stretch here</name>

<description> Enter the Stretches instructions </description>
 <URL name="the stretch name ">/Stretches/Graphics/Name of
the stretch image.jpg</URL>
</stretch>
```

3) Enter the areas that the stretch you adding plans to benefit out of the following: Leg, Hands, Eyes, Back or Neck. Enter this between the <type> and </type>tags.
4) Enter the name of the stretch you are adding between the <name> and </name> tags.
5) Enter the instruction of how to perform your stretch between the <description> and </description> tags.
6) In the URL name tag "<URL name="the stretch name ">" enter the name of your stretch in place of "the stretch name ".
7) Place the image of you stretch in the Graphics folder. This folder is a sub folder of the Stretches folder, which is inside the Aries folder.
8) Change the URL in the xml to match that of your image. This can be achieved by changing the "Name of the stretch image.jpg" in the **/Stretches/Graphics/**Name of the stretch image.jpg. An example of this would be /Stretches/Graphics/Stretch2.jpg
9) Save the XML file.
10) Re-build the Aries project (see Section 2)

6.2. Add another Scheduled Break Screen

In order to add another scheduled break to the system the developer must do the following:

Difficulty: Moderate

1) Add three JCombo Boxes and a TextLabel to the Setting tab's pane on Aries GUI (illustrated below in figure 1)

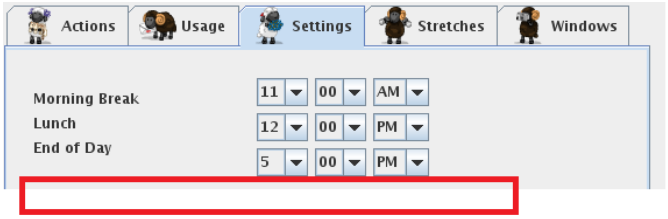


Figure 73 - Proposed placement of new Schedule break

2) Enter the desired text into the Text Label
3) Put values from 1 to 12 into the Hour ComboBox
4) Put the values 00, 15, 30 and 45 into the Minute ComboBox
5) Put the values AM and PM into the AMPM ComboBox
6) Now, Open the getSettings class and enter an If statement for your new break. Like

```
 if (controller.GetNewBreakAMPMValue().equals("PM")) {
(see other loops of examples)
```

7) Then set the time to twenty four hour

```
String realNewBreakHour = Integer.toString(NewBreak24);
```

8) Then put the time into the correct format. Like shown here:

```
NewBreakTime = "" + realNewBreakHour + ":" +
controller.GetNewBreakMinuteValue() + ":00";
```

9) Next, open the GuiWrapper Class
10) Insert getter methods for each of the three Combo boxes that you added to the GUI. An example is shown below:

```
public String GetNewBreakAMPMValue() {
    return (String) NewBreakAmPmCombo.getSelectedItem();
}
```

11) Next Open the DisplayTime class and insert the following if else clause to the if statement located at the foot of the class. An example is shown here:

```
} else if (timeNow.equals(getSettings.NewBreakTime)) {
    scheduledBreakType = "New Break";
    controller.newDisplayScheduledBreak();
```

12) Lastly, Open the displayScheduledBreak class and add and else if statement to the if clause like shown here:

```
}else if(Displaytime.scheduledBreakType.equals("New Break")){
  //Creates an icon to display on the break screen
  ImageIcon icon = createImageIcon("Path to the breaks image");
  //Sets the New Break and displays it on the break screen
  Bcontroller.SetBreakName("New Break");
  //display the icon on the screen
  Bcontroller.SetBreakImage(icon);
```

Change the `"New Break"` to the name of whatever the break it is you want to display and change the `"Path of the breaks image"` to an image in the /Graphics folder inside the Aries folder. You can add any image you want to this folder.

Appendix C
Installing Java SE 6

Linux

Java SE 6 is provided for Linux by Sun Microsystems.
For system requirements please see:
http://www.java.com/en/download/help/6000011000.xml
All major Linux distributions support Java SE 6.
To install Java for Linux, please see:
http://www.java.com/en/download/help/linux_install.xml
For Fedora and Red Hat based systems an RPM installer is available and can be found here.
http://www.java.com/en/download/help/5000012000.xml
For Debian based distributions, such as Ubuntu, Java SE 6 can be installed through the Package Manager.

Windows

Java SE 6 is provided for Windows by Sun Microsystems.
Java SE 6 is supported on the following Windows versions: 2000(Service Pack 4), XP (Service Pack 2), Server 2003, Vista and Windows 7.
Java SE 6 is available on all 32bit versions of these systems and is supported on 64bit versions of XP, Server 2003, Vista and Windows 7.
For system requirements please see:
http://www.java.com/en/download/help/6000011000.xml
To install Java SE 6 on Windows, see:
http://www.java.com/en/download/help/windows_manual_download.xml and follow the instructions.

Further Questions

For any further information about installing Java SE 6 please see http://www.java.com/en/download/help/index_installing.xml and choose which platform it is that you are running from the drop down menu located in the top right.

Appendix D
RSI Physical solutions, Techniques and Surgery

1. Breaks

The majority of professionals and experts in the field of ergonomics and RSI agree that taking breaks during typing is vital technique for preventing and recovering from RSI. Below is a brief overview of the three sorts of breaks that users should participate in:

- **Eye breaks**: The constant use of computers means that the user is focusing on screen for a long time, this causes the user to blink less often which in turn exposes the eye to air for a longer period of time. This can result in changes of how the eye usually functions. In order to prevent this from occurring it is recommended that every 15 minutes the user should look away from their screen and focus on a distant object more than 20 feet away for a period of time ranging from 30 seconds to 2 minutes, depending on their eye strain. To prevent eye dryness which aides strain, it is recommended that the user blink their eyes, rapidly for 10 seconds, every 30 minutes. This effectively clears dust from the surface of the eye and refreshes the eye's tear film.

- **Micro-breaks:** The observation of typists has revealed that typing is not a continuous task, but instead occurs in short bursts. Whilst typing it is important to break up the monotonous, repetitiveness of the activity by taking a brief break during these bursts of activity. A micro-break is a break designed to interrupt the repetitive nature of your activities rather than prevent them. In the prevention of RSI these micro breaks are of vital importance.

- **Rest-breaks:** In order to help prevent RSI, breaks should be taken every 60 – 90 minutes. During these breaks it is recommended that the user gets up and moves around. This can be achieved by getting a glass of water or a coffee for example. Doing small exercises during these breaks is also recommended in order to relieve any muscle fatigue. Just by changing what it is that the user is doing will allow other muscles to function and this will relieve fatigue and aid in stamina.
  For users that already experience RSI related pains it is important that the get the duration between the breaks correct. It is generally accepted that the period of time between breaks in this case is the time it takes till you start to experience discomfort, minus 10 minutes.

2. Ergonomics

One of the biggest contributing factors to the development of RSI is bad, static posture. Bad posture combined with spending long periods of time with your arms flexed and without taking appropriate breaks can lead to increased static loading across many muscles throughout your body. These muscles can then become shorter in length and denser over time resulting in impeded blood circulation in the shoulders, neck and arms.

RSI can occur long before you will feel and physical discomfort. The first signs of RSI in which you feel pain can become visible up to four to five years before you may experience any debilitating hand pain. The symptoms to look out for are any back pains, shoulder pains and/or neck aches that many experience on a regular basis whilst using your computer. These pains can lead to a tethering of the nerves within the areas of discomfort. After several years of performing this activity, this tensed state becomes the default state and RSI ensues. However by correcting the ergonomics of your work practise, taking appropriate breaks and performing correct stretches, one can prevent and recover from RSI.

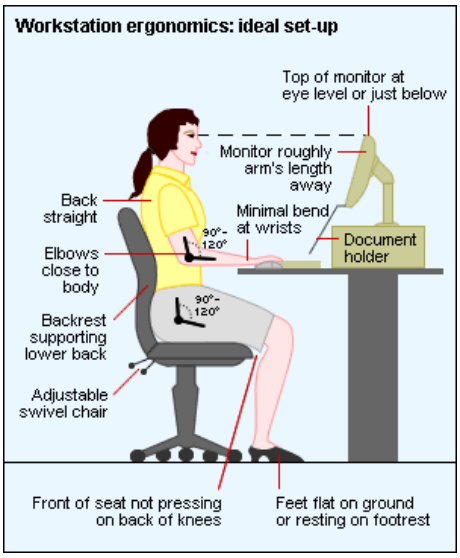

Workstation ergonomics: ideal set-up
Top of monitor at eye level or just below
Monitor roughly arm's length away
Back straight
Minimal bend at wrists
Document holder
90°-120°
Elbows close to body
90°-120°
Backrest supporting lower back
Adjustable swivel chair
Front of seat not pressing on back of knees
Feet flat on ground or resting on footrest

Figure 1.0.1 - Ideal Ergonomic Workstation Set-up

3. Posture

A constant source in the development of RSI's is that of poor posture. The natural muscular balance of your body is disrupted by poor posture and after time, your body begins to compensate for this imbalance by contracting nearby muscles for additional support, tightening the connective tissues to further support the imbalance. Over a period of months and years if unresolved, this tightening of tissues can result in the body becoming rigid, stiff and may also interfere with the normal contractions/relaxations of muscles which inhibit the movement of surrounding nerves and blood vessels. Correcting poor posture is imperative in recovering from RSI. The ideal set up is shown above in figure 1.0.1 and below in figure 2.0.1 and 2.0.2.

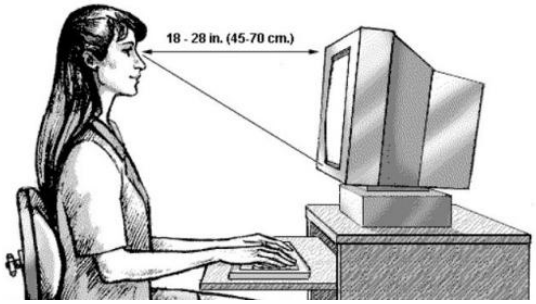


Figure 2.0.1 - Correct Monitor Height

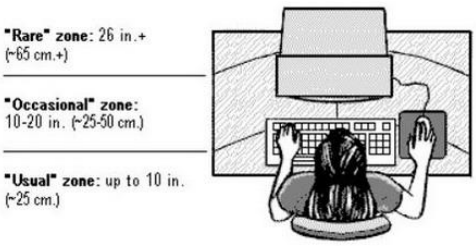


Figure 2.0.2 - Correct Desk Distances

4. Lighting

The main aim to prevent RSI is for the user to be as comfortable as possible. Bright light in a workplace can be very irritating so it is important to make sure the lighting isn't too bright and that if you have glare on your screen, that you move the screen, lower the light level or use a good quality anti-glare screen. Also be sure not to have a bright light, such as a window, directly behind or in front of your monitor as this variation of light can be a cause of eye strain.

5. Ventilation

Whilst working, it is important that both you, and your computer, have adequate fresh air. Stuffy, poor ventilated rooms are the cause of headaches. To prevent RSI it is important that your working environment is comfortable.

6. Noise

Loud and unnecessary noise can cause stress which in turn causes muscles to tense which increases the risk of RSI. Your working environment should be quiet or have low levels of background noise.

7. Mouse Techniques

The first and most effective technique to lessen mouse strain is to try and avoid the mouse whenever possible. Nearly all tasks can be done using the keyboard instead of the mouse and the keyboard allows you to distribute the necessary workload over both hands. Using hotkeys and key commands not only lessens your muscle workload but improves your efficiency too.
The following techniques significantly lessen the strain when using a mouse:

- Avoid resting your wrist and forearms when moving the mouse.
- Move the mouse from your shoulder, not from your wrist. (see fig 6.0.1)

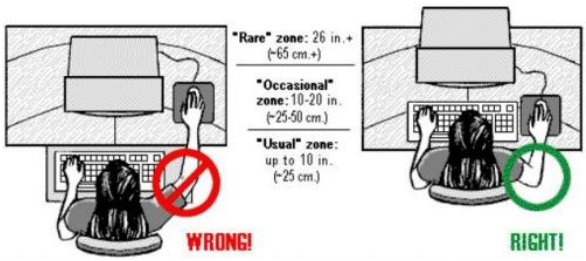


Figure 6.0.1 - Correct Mouse Action

- Keep a straight line from your forearm through the mouse. (illustrated in fig 6.0.2)
- Avoid gripping the mouse tightly, pinching the mouse; instead hold the mouse gently with all your fingers.
- Avoid lifting your smallest finger when using the mouse.
- Click the mouse buttons gently and patiently.

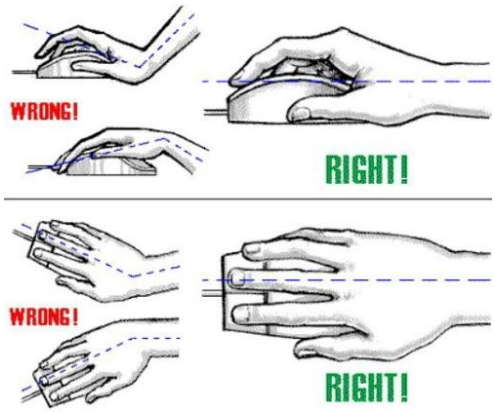


Figure 6.0.2 - Correct Mouse Angle

Ergonomic software is another option to prevent RSI when using a mouse. This is because working at a computer can become hypnotic, and often you don't realise just how long you have been working and how much you have been typing and using the mouse. The ergonomics software package will run in the background and monitor how much you have been using the computer and then will prompt you to take a rest break at appropriate intervals throughout the day or during prolonged periods of activity. It should also suggest simple exercises for you to do during this break.

Another feature that can be offered to the user is an AutoClick feature. This can dramatically reduce the amount of strain experienced by using a mouse and has no significant productivity loss. This feature works by counting every time the mouse stops as a click. This could be a single, double or right click depending upon your settings. Although this feature can take bit of getting used to it is an effective way of decreasing RSI when developed properly.

AutoClick is particularly useful for Carpal Tunnel Syndrome and de Quervain syndrome sufferers, and for its prevention. De Quervain's syndrome is a painful condition that affects tendons where they run through a tunnel on the thumb side of the wrist. (Illustrated in figure 6.0.3 and 6.0.4 below)

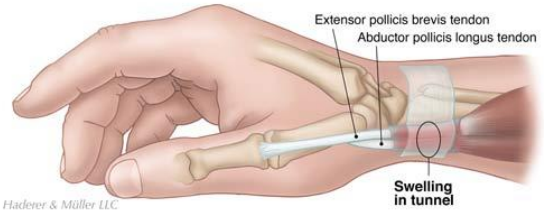


Figure 6.0.3 - De Quervain's Tenosynovitis

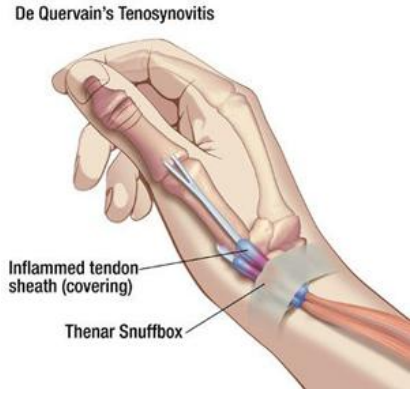


Figure 6.0.4 - De Quervain's Tenosynovitis

8. Typing Techniques

Being able to correctly use a keyboard is of vital importance in the war against RSI. If you are not using your keyboard correctly whilst typing then your incorrect typing techniques can lead to and worsen RSI. Your wrists should be straight as shown below in Fig 7.0.1:

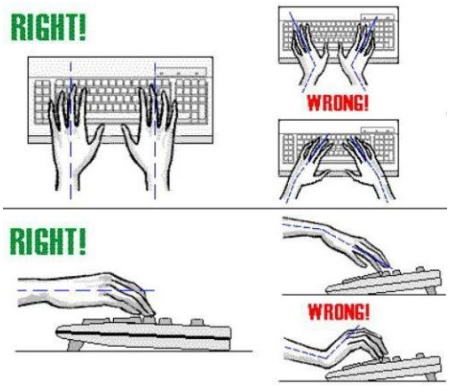


Figure 7.0.1 - Correct Typing Posture

8.1 DVORAK Layout

If you suffer from RSI whilst typing then it may be a good idea to switch to a DVORAK keyboard layout as it relieves the hand action during typing. It may come as a shock to you that the more common QWERTY layout was specifically designed to be most inefficient and to maximize hand movements. The reasoning behind this was that older typewriters used to get jammed by touch-typists. Regrettably this convention has continued into the modern information age.

Fortunately, during the 1920s and 30s two Doctors, Dr. August Dvorak and Dr. William Dealey, designed an alternative layout to minimize hand movements and improve efficiency. This can design is shown below in Fig 7.1.1.

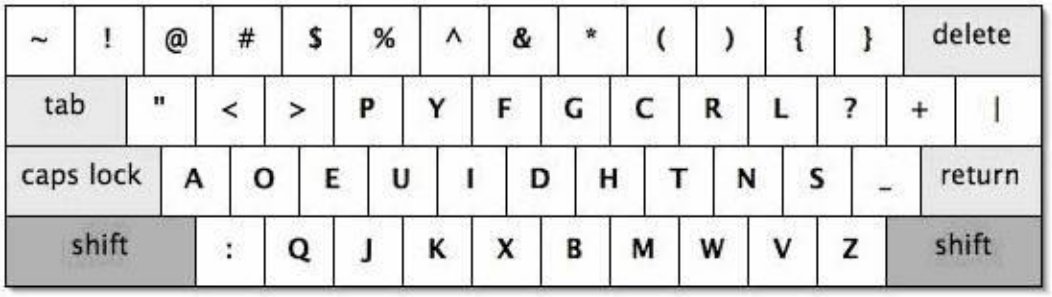


Figure 7.1.1 - DVORAK Keyboard Layout

Dvorak and Dealey studied letter frequencies and the physiology of the hand and created the layout to adhere to these principles:

- It is easier to type letters alternating between hands.
- For maximum speed and efficiency, the most common letters and digraphs should be the easiest to type. This means that they should be where the fingers rest, which is called the Home row.
- Likewise, the least common letters should be on the bottom row, which is the hardest row to reach.
- The right hand should do more of the typing, as the majority of people are right-handed.
- It is more difficult to type digraphs with adjacent fingers than non-adjacent fingers.
- Stroking of the keys should generally move from the edges of the board towards the middle (as an example, rap your fingers on a table and see which is easier: going from little finger to index or vice versa). This motion on a keyboard is called inboard stroke flow.

Dvorak typists tend to be much faster than Qwerty typists, both because of ease of typing (70% of keying is on the home row, versus 32% on Qwerty) and because of the reduced error rate on Dvorak due to its ergonomics, but the ease in learning and typing is its greatest advantage.
According to people who suffer from RSI, the DVORAK layout really does offer a benefit as it evenly distributes the typing load between both hands and helps to reduce the overall typing action. Although this is not a cure to RSI, by switching to a DVORAK keyboard layout is just another good way to help aid the recovery of RSI. This method must be combined with other forms of therapies.

9. Surgery

In extreme cases RSI can lead to the need of surgery. This is only recommended, if at all, if all other forms of non-surgical treatment have failed and if you have suffered from a musculoskeletal disorder for a long period of time. If your doctor detects a wasting of the muscles or loss of sensation due to computer activity surgery may again be suggested.

Hand surgery can be performed using a few methods, and the few RSI-related disorders which have surgical solutions are:

- Carpal Tunnel Syndrome
- Trigger Fingers
- Arthritis

Surgical solutions for these disorders use three techniques:

- Carpal Ligament Release Surgery
- Triggers Fingers Release Surgery
- Arthritis Surgery

Below is a brief description of the three surgeries.

9.1 Carpal Ligament Release Surgery

The goal in carpal ligament release surgery is to cut in two the flexor rentinaulum ligament which is located in the wrist. This wide ligament runs across the hand, from the scaphoid bone to the hamate bone and pisiform. The ligament forms the roof of the carpal tunnel and once cut (in line with the ring finger) the pressure is released on the median nerve underneath it. After surgery a lengthy physiotherapy program is needed. This procedure is shown in Figure 8.1.1 below.

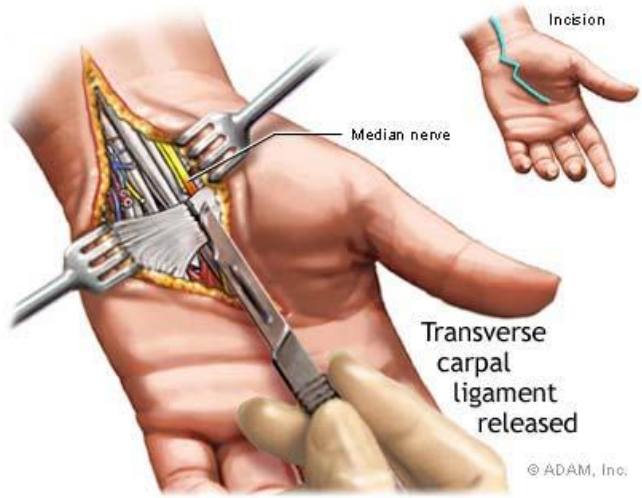


Figure 8.1.1 - Transverse Carpal Ligament Surgery

9.2 Trigger Finger Release Surgery

Trigger finger occurs when a tendon in your finger, one that allows your finger to bend, becomes swollen and enflamed. This then catches on the opening of the sheath that it passes through at the base of your finger. The result of this is that either your finger makes a clicking sound or your finger may become locked in the closed position.
During surgery the surgeon will make a small incision into your palm at the base of the affected finger. They will then enlarge the part of the tunnel that is currently restricting the

enflamed tendon. After the tunnel has been enlarged the tendon should be able to function normally.

9.3 Arthritis Surgery

In this type of surgery, the arithiel surfaces at the joints are removed or sanded away, and a material is inserted between the two ends of the bones. This material may consist of a natural tendon taken from the patient or may be a synthetic plastic rubber shaped to fit into the joint. Some surgeons may simply choose to fuse the two bones making up the joint together thus eliminating the joint and pain.

9.4 Surgery Success Rate

Surgery for RSI has no proven track record. Indeed it has been reported of having a 57% failure rate. Many of the people who go through with surgery report experiencing no benefit as a result of surgery or reporting having a later relapse; which may cause permanent tissue change. A study by the American Medical Association reported that intermittent pain was reported by 42%, digital numbness by 32% and tingling highly 35% of people who underwent surgery. As a result of surgery for "carpal tunnel syndrome", a severed carpal ligament (flexor retinaculum) can also lead to reduced grip strength. Surgery for Carpal Tunnel Syndrome (CTS) is the second most common type of surgery in the US, with well over 230,000 procedures performed annually. With more than eight million people reported affected by carpal tunnel syndrome each year. Surgery for carpal tunnel syndrome is the second most common type of musculoskeletal surgery, with back surgery being the most common.

According to the Bureau of Labour and Statistics and the National Institute for Occupational Safety and Health (1997-2000 Statistics) only 23% of patients who have under gone carpal tunnel syndrome have returned to their previous professions.

If you have an RSI besides CTS, surgery tends to offer no benefits for the patients and I would not recommend it.

9.5 Cost of surgery

According to the Agency for Health Care Policy and Research in the U.S. musculoskeletal disorders are the U.S.'s most costly category of workplace injuries and illnesses. In addition to spending $20 billion annually on workers' compensation costs due to RSIs, the U.S. spends another $100 billion on lost productivity, employee turnover, and other indirect expenses. Current estimates suggest that repetitive strain injury conditions cost the UK industry between £5 billion and £20 billion every year.

According to a survey conducted in 1998 by CTD News in the U.S. a common office RSI, such carpal tunnel syndrome, averaged at a cost of $1,918 per case in 1997. However this does not include surgery or rehabilitation as the national average is more than $12,000 per hand. The average lifetime cost of carpal tunnel syndrome, endured by employees, including medical bills and lost time from work, is estimated to be about $30,000 for each injured worker.

Although surgery may seem to be an attractive option, easy option to some, it is certainly not the most effective one. The numbers certainly go against those of the surgical techniques discussed above with the cost of both the diagnosis and surgery being very high. Non-surgical techniques should be considered and explored more rigorously before

resorting to surgery which can only be considered as a last resort. The best and most effective way of fighting RSI is to try and prevent it in the first place.

Appendix E
Architecture Research

Contents

Technical Architecture 2
System Architecture 2
Application Architecture 2
Enterprise Architecture 3

1. Architecture Research

It was important to consider all the various types of architecture that the IASA Glossary[1] lists before deciding upon the correct one for this system.

IASA architecture types:

- Application
- Business
- Data
- Enterprise
- Hardware
- Information
- Infrastructure
- Network
- Platform
- Performance
- Security
- Software
- Solution
- System
- Technical
- Technology
- Web

We will not delve too extensively into all of these types as many overlap greatly, create cyclic dependencies, are subjective in places and many refer to each other. With so many different terms out there, each overlapping, it is fair to say that many in the industry are confused as which model to apply where. The IASA terms offer an insight as to which should be applied where but as of yet offer no widespread standard for the discipline and all the terms can be a bit overwhelming. Several forms of "architecture" are commonly referred to within the field of software development. It is these that I will look into with further detail to see which best applies to my system.

[1] http://www.iasahome.org/web/home/ITarchitecture/Glossary - The International Association of Software Architects.

1.2. Technical Architecture

The term “technical architecture” is commonly a phrase coined when first attempting to describe a system’s architecture but it does not closely tie to any type of software. It is this inability to define a specific system that renders this form of architecture to be particularly meaningful when describing a system layout design.

A technical architecture is formally used to refer to any of the architectures a system may have and is often used to refer to a collection of architectures that are defined within a system. It is arguably a more general term than “system architecture”.

1.3 System Architecture

System architecture refers to the way in which the desired functionality is fulfilled by hardware and software components, as well as how these components interact with one another and the intended users of the system. Several different business functions can be encompassed under the term of system architecture.

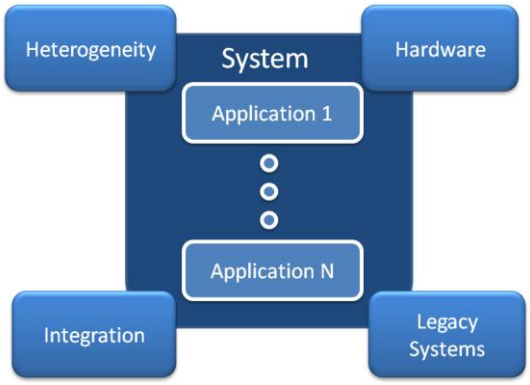


Figure 74 - System Architecture

The system architecture naturally focuses upon the system-wide designs decisions and similarities as it deals with the system as a whole. Although it needs to consider the implications of such decisions for individual functions it may not resolve them fully within each and wait till the application architecture which relates to each function and where the issue can be better addressed.

1.4 Application Architecture

Application architecture is the subset of system architecture. The business function often determines the scope of the application architecture unlike in the systems architecture.

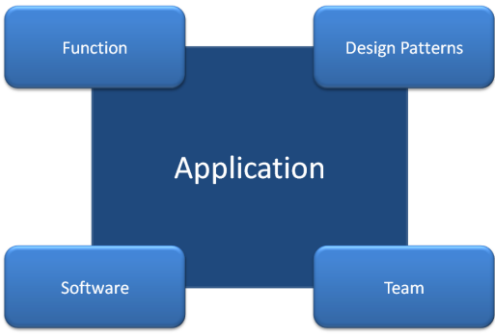


Figure 75 - Application Architecture

It is typical for the application architecture to be defined to a lower level than that of the system architecture. This is mainly due to the fact that it needs to refine the system architecture to provide the design decisions that relate exclusively to the business function rather than to the whole of the system.

1.5 Enterprise Architecture

The term enterprise architecture is one that is often mistakenly used by people that work on enterprise systems or systems that involve components that are touted as enterprise level. This form of architecture however is more involved with the mapping of business process and the needs of technical capabilities of the organisation, which include personnel, strategy, distribution and how the business' changing needs will be addressed.

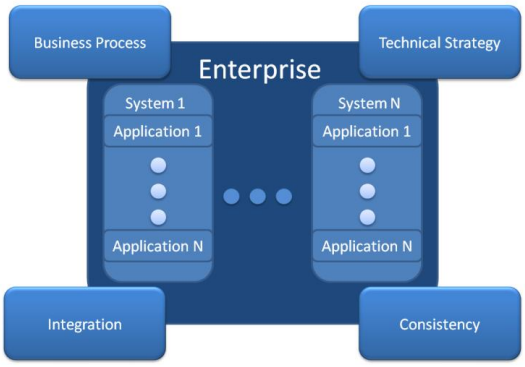


Figure 76 - Enterprise Architecture

The enterprise architecture therefore encompasses the whole development spectrum, being enterprise wide and requires careful inspection of all the business functions and their strategic requirements. Enterprise architectures should be treated with a pinch of salt however as they are often mistakenly used to represent some form of enterprise software.

Appendix F
Focus Group Plan

1. Introduction

This study will be carried out by Peter Kindness as part of his research towards his 4th year honours project, which looks into the causes of RSI with a view to reducing or eradicating these causes.

Participation in this study is completely voluntary.

The session will be recorded in print as to allow further analysis of answers to be carried out at a later date.

2. Discussion Guidelines:

The aim of this discussion is to be informal, so there is no need to wait for me to call on you to respond. I encourage you to respond directly to comments made by the other members of the group as they respond. If you do not fully understand one of the questions then please do not hesitate in letting me know. I am here to ask questions, listen and make sure everyone has a chance to share his or her view.
If you seem to be struggling on a tropic, I may interrupt you and if you if you have yet to respond I may call on you directly. This is in order to make sure that I obtain everyone's perspective and opinion.
Everyone's identities, participation and remarks will be kept private, so I encourage you to speak freely and honestly

3. Introductory Questions

Although all members of the focus group know one and other, the following introductory questions were asked to both introduce myself to the group and learn about them and also to just break the ice and get the discussion started.

Get all participants to introduce themselves:

Name
Job Title

4. Knowledge of RSI

Ask the participants whether they know what RSI is and where they have had experienced any RSI related symptoms before, whether this is in the workplace or not.

Invite discussion on any types of symptoms mentioned.

5. Current solutions to RSI

Ask the participants whether they currently use any solutions to RSI.
Query whether these are hardware solutions, such as wrist pads, or whether they are software solutions, such as the proposed system Aries.

Establish which methods are most popular and whether they work or not.

Discuss the merits and flaws of any systems mentioned.

6. Aries

Introduce the group to the concept of Aries. Ask them whether they think they would benefit from such a system.

Explain Aries' preliminary features and gather their thoughts on these proposed features.

Ask whether they feel any features are currently missing that could be helpful.

7. Closure of session

Ask the group if they have any additional thoughts or suggestions. Give a brief overview of the group results.

Thank everyone for their time and adjourn the meeting.

Appendix G
Focus Group Questions

Group: Hampton Associates Web Development Team

1. General RSI Related Questions

Q1) Do you know what RSI is?

The group know what RSI was as they work with computers all the time.

Q2) Have any of you ever experienced any problems relating to RSI?

One member of the team suffered from a sore pinkie (small finger) on his left hand, due to constant pressing of the shift key. This may be due to the lack of a properly ergonomic keyboard
One of the members of the team had had RSI related problems but though other activities rather than computers. In this case it was tennis elbow, he believed was due to playing golf.

Q3) When you experienced these problems, how did you look to resolve them?

Finger - they didn't look to actively resolve the problem and the syndromes subsided after a period of time.

The person who suffered from other activities rather than computers, (golf) bandaged his elbow and took a break from playing the sport.

Q4) Are any of these syndromes recurring?

No.

Q5) When and how did you first become aware of RSI?

Through media reports (news), wrist pains were highlighted due to the increase of typing with keyboards. They all became aware of RSI a long time ago.

Q6) Are you aware of any RSI prevention system out there at the moment?

No software, only hardware; such as ergonomic keyboards and wrist protections and ergonomic mice.

2. Software Currently Used and Daily Routine

Q7) Which operating systems do you use?

Windows (Vista and 7) – rest of the studio (other departments) use Macs

Q8) What programs do you use most in your daily routine?

Web browsers, Text editors (Word, HTML text editors) and Photoshop.

Q9) Do you think that an RSI prevention program is a one that could help?

Yes.

Q10) Do you think such a system would be a hindrance to your work?

Depends on how strict and encroaching it is upon their work.

Q11) How frequently do you take breaks throughout your working day?

Hourly

Q12) How long do you think the breaks should last for?

5 minutes

---

3. Evaluation of "RSI Prevent" proposed features

Q13) Do you think it would be better to take shorter more frequent breaks, or longer less frequent breaks?

Longer less frequent breaks as the short ones may getting annoying and restrict the flow of work.

Q14) Would you be willing to use a system like "RSI Prevent"?

Yes, as long as you can override it. If it locks you out of the system when you really need to achieve something then that is no good.

Q15) How would you feel about performing various RSI preventing stretches in the workplace?

If they are not attention drawing and do not distract others from their work – fine.

If they are noisy then no. (such as star-jumps)

Q16) Do you think the system should have sounds, or do you see this as being distracting to other people?

All members would turn the sounds off if there were sounds.

Q17) Do you think the system should record statistics about your work pattern?

Yes. As it would be quite interesting to look at what you have achieved.

Q18) How long do you think the period of time between the breaks should be?

The times ranged from 45 minutes to one and a half hours.

Q19) Would you be interested in the system monitoring the use of certain programs such as Word, Excel etc?

Yes, as this would be interesting to see at the end of the day.

Q20) Can you think of any other features that would find helpful in such a system?

If you could tie in a time keeping device.

For example in web development they need to track the number of hours spent on certain projects.

A prompt when you open the web developer to select the project you are working on. And then start a timer.

Have a list of the project you are currently working on which can be deleted once that certain project is completed.

This would be a very helpful feature.

Appendix H
Questionnaire

As part of my Honours Project, I am designing and implementing a system to help prevent Repetitive Strain Injury (RSI). RSI is a pain syndrome affecting millions of people world-wide and is the number one occupational hazard in the entire US. This questionnaire is designed to get an insight into how people currently use their computers and as to what habits they currently process.

How old are you?
□ Younger than 16 □ 16 – 24 □ 25 – 34 □ 35 – 44 □ 45 – 54 □ 55+
What gender are you?
□ Male □ Female □ Do not wish to disclose

Occupation: _____

How many hours a day on average do you use a computer?
□ 1 or less □ 2 - 3 □ 4 - 6 □ 6 – 8 □ 8+
Have you ever had problems relating to RSI (Repetitive stress injuries)?
□ Yes □ No □ Do not know what RSI is
What do you mostly use your computer to do?
□Typing □Internet browsing □Spreadsheets □Monitoring Systems
□Emails □Gaming □Other (please specify)_____
Where is it that you mostly use a computer?
□ Work □ Home/Leisure

Are you aware of any RSI prevention products currently available? (Tick all that apply)
□ Not aware of any □ Workpace □ RSI warrior □ RSI Screensaver
□ EyeDefender □ Stress Buster □ Break Time □ Break Reminder
□ Albion StopNow! □ Workplace Angel □ Other (please specify)
_____
How often do you take a break during a working day?
_____

Which of these features in the system would you prefer?
□ More frequent, shorter breaks □ Less frequent, longer breaks

Is your workstation well designed for your job? □Yes □No
If not, why not?
_____

Can you think of any obvious & immediate improvements that could be made to your workstation?
_____

_____
_____

Does the machine/process-line/management/etc determine the speed of your work or can you control it? Please choose an appropriate value, 1 being the least and 5 being the most.

1	2	3	4	5
o	o	o	o	o

Entirely dictated by you Entirely dictated to you

Is your output measured each day/is there a monitoring system in operation?

_____
_____
_____

Other Comments_____
_____
_____
_____
_____
_____

Thank you for taking the time to complete this questionnaire.
All data obtained will be completely anonymous and will not be passed onto any third parties.
Any further questions, please don't hesitate to email me on *p.kindness.06@aberdeen.ac.uk*